AFIT/GOR/ENG/98M-01

AN ADVANCED VISUALIZATION METHOD FOR AN OPERATIONS RESEARCH ANALYSIS

Captain Steven C. Oimoen

AFIT/GOR/ENG/98M-01

AFIT/GOR/ENG/98M-01

AN ADVANCED VISUALIZATION METHOD

FOR AN OPERATIONS RESEARCH ANALYSIS

THESIS

Captain Steven C. Oimoen

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Member

Keith Shomper
Member

Thomas C. Hartrum
Chairman

Acknowledgments

I would first and foremost like to thank the greatest contributor, my Lord and Savior, Jesus Christ. Without his strength, guidance, wisdom, and patience, I would have been unable to accomplish this thesis effort. The Lord encouraged me when I doubted myself, provided solutions when I could see none, and strengthened me with a purpose for my life and studies.

My advisor, Dr. Thomas Hartrum not only kept me on track throughout this project, but provided me the freedom and flexibility to complete my studies. I appreciate his willingness and patience to advise a student outside of his department. I would also like to thank LTC Jack Kloeber, my operational sciences representative and co-advisor, for his enthusiasm, experience, and guidance.

In addition, I would like to thank the following individuals: Craig Gorby and Dale Robinson, my friends, who have always believed in me; Ellen Morton, for her prayers and encouragement; my miniature schnauzer, Sally, for taking long walks with me as I pondered thesis strategies; and my children, Brian, Eric, and Joseph - my thoughts are always with you. Finally, a special thank you to my sister, Kerri, who has always supported me beyond belief - thanks Koo!

Table of Contents

List of Figures

List of Tables

Abstract

Visualizing multidimensional data using only two dimensions and conventional visualization techniques limits the understanding of the data set. Underlying structures or patterns within the data can easily go unnoticed. In order to gain additional insight into an analysis, incorporation of visualization and multidimensional graphics into the analysis results should be accomplished. The results must ensure that the information portrayed is not misleading or misunderstood. The integrity of the data must be preserved throughout the transformation. The primary objective of this research effort is to identify techniques to visualize multidimensional data and then develop a software tool to display the multidimensional data in order to gain insight into the data and analysis performed.

AN ADVANCED VISUALIZATION METHOD FOR AN OPERATIONS RESEARCH ANALYSIS

1. Introduction

1.1 General Issue

People observe and draw conclusions, whether consciously or unconsciously, from what they see. It is a stated truism that "seeing is believing". However, seeing is not just believing. To see may also be to understand, provided what is being looked at has the information within it to aid the understanding process.

Throughout recent history, scientists and engineers have used graphs and charts to represent their data visually or have just looked at the numbers trying to understand the phenomena that the data set represented. An important aspect of an operations research analysis should be to visually represent the data (Deckro, 1998). Because the real world is richly multivariate, the data that is to be displayed is often multivariate as well. With modern day computer and display technology, it has become possible to represent data visually in two or higher dimensions using a variety of techniques and methods. This thesis develops an approach to visualize multidimensional data associated with the value hierarchy model for the Paducah Gaseous Diffusion Plant Waste Area Group (WAG) 6.

1.2 Background

The Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA) and the National Contingency Plan (NCP) establish criteria for determining

the appropriate environmental response by outlining the procedures to be followed in performing cleanups, remedial actions, or removals. The Paducah WAG 6 value hierarchy used value-focused thinking and multiple attribute preference theory techniques (Kerchus, 1997: 2-10) to produce a CERCLA-based decision analysis model to aid site decision makers in selecting a remediation strategy (Kerchus, 1997: 3-7). The model ranked 28 specific remediation strategies based on their potential to meet CERCLA's five balancing criteria: implementability; short-term effectiveness; long-term effectiveness and permanence; reduction of toxicity, mobility, or volume through treatment; and cost. The data hierarchy consisted of the five CERCLA balancing criteria that were decomposed and quantified by 28 evaluation measures. Single dimensional value functions were developed for each evaluation measure. Weights were assessed for all tiers within the value hierarchy. The CERCLA value model provides decision tools that can help site decision makers make a better informed, more defensible decision. Kerchus provides a complete description of the value model and its derivation (Kerchus, 1997).

The software application used in the visual analysis of the Paducah WAG 6 CERCLA-based value model provided graphical representations of the scores for each of the five balancing criteria, but was somewhat limited in its ability to visually represent multidimensional data and convey the proper insights to the decision makers. The visual representations were basically limited to displays in two-dimensions with little creativity in the use of space and motion. Tufte states that "escaping flatland is the essential task of envisioning information" (Tufte, 1991: 12). Inappropriately limited displays can diminish the effectiveness of an operations research analysis.

1.3 Objective

This research effort focuses on identifying techniques to visualize multidimensional data and then developing a software tool that utilizes these techniques to gain insight into the data and make better, more informed decisions. Visualization in scientific computing continues to emerge as a major computer-based field. As a tool for applying computers to science, visualization in scientific computing promises radical improvements in human/computer interaction (Gershon, 1993: 343). Recent work in the field of scientific visualization demonstrates how computers can serve as an intermediary in the process of rapid assimilation of information (Robertson, 1993: 65). Large sets of data, when transformed into a graphical form, are reduced in such a way that human perception can detect patterns, previously unnoticed within the data set. Information visualization attempts to display structural relationships that would otherwise be more difficult to detect.

Tim Bartel, a researcher at Sandia National Laboratories, used visualization to develop a boat which the United States Coast Guard can safely drop into the water, complete with crew, from a helicopter. Bartel was quoted in Weber's article as saying "without this kind of visualization tool, the task of analyzing the huge amounts of data from the simulation would be almost impossible" (Weber, 1993: 121).

The issue then is how to translate multidimensional data onto a two-dimensional plane (computer screen) to convey results and communicate insights. Creating the ability to view the data as multidimensional objects, exhibiting their own characteristics, should improve the understanding of and insights gained from an operations research analysis. For instance, visualization techniques can be applied to the data, not only to view, but to

perform sensitivity analysis as well. The results obtained will enable a decision maker to actually see how the changes may impact the results.

1.4 Statement of the Problem

Visualizing multidimensional data using only two dimensions and conventional visualization techniques limits the understanding of the data set. Underlying structures or patterns within the data can easily go unnoticed. In order to gain additional insight into the analysis of the Paducah WAG 6 CERCLA-based value model, incorporation of visualization and multidimensional graphics into the analysis results will be accomplished. The results must ensure that the information portrayed is not misleading and that the integrity of the original data set be preserved throughout the transformation.

1.5 Research Approach

This research examines methods to visualize multidimensional data. The literature review in Chapter 2 discusses information visualization as an emerging technology. The research presented in the literature review was to determine the best visualization approaches to display multidimensional data. The researcher surveyed a currently used software application, Logical Decisions for Windows (LDW), which is used in multiattribute operations research analyses (Logical Decisions, 1995). The research also determined the shortcomings of the currently available software application concerning its ability to visualize multidimensional data.

1.5.1 Software Approach

One approach to view data as multidimensional objects is the use of computer graphics and visualization. Computer graphics and visualization have revolutionized the way we are able to interact with and understand data. Transforming our data into information provides for a better analysis and more importantly a better understanding of that information to the decision maker.

Visualization tools employ computer graphics and are making a major impact in this particular area of study. The graphics tools offer utilities for analyzing large and complex data sets and pinpointing areas of significance in multivariate data. The goals of good visualization are to gain insight and understanding of the data through the creation of computer graphics images, and to communicate this information to others.

With respect to the software visualization tool, a requirements analysis was performed. Potential users of the software visualization tool, as well as thesis committee members, determined the functional requirements necessary. Based on the approved functional requirements of the software visualization tool and the visualization techniques selected to view multidimensional data, a high-level and detailed design of the software visualization tool was constructed. During the design phase, the programming language for the software visualization tool was addressed. Once the designs were approved and the programming language determined, the coding phase began. Upon completion of the coding phase, the software visualization tool was tested and debugged. After the software had successfully passed the testing phase, the software visualization tool was validated.

1.5.2 Validation Process

The validation process consisted of several Air Force officers, who are currently graduate students in the Operations Research program, comparing the visual outputs of Logical Decisions for Windows with those of the newly created software visualization tool (Appendix C). An analysis of the validation test results was performed. Chapter 3 explains the validation process and Chapter 5 provides the results of the analysis.

1.6 Assumptions

In order to compare the different software displays and validate the findings, data sets were required from the Paducah WAG 6 CERCLA-based value model. Also, sufficient "experts" were needed to evaluate the visual displays of the data.

1.7 Scope/Limitations

The software visualization tool presents the information in a multidimensional visual representation, displaying data from the Paducah WAG 6 CERCLA-based value model. The software visualization tool's platform is a personal computer (PC), with a Pentium based processor, running the Windows 95 operating system and Microsoft's Internet Explorer 4.0. The emphasis of this software visualization tool is on the visual techniques used to display multidimensional data and not the software visualization tool's user interface.

1.8 Summary

This chapter presents a background and basic introduction of the problem to be addressed. Chapter II is the literature review and provides a more detailed background on

specific topics with respect to visualization of data. Visualization examples, general visualization guidelines, applied visualization techniques, and multivariate visualization techniques are presented. Chapter III explains the methodology of how to visualize data in depth. It also discusses the experimental procedure used and how the data was collected and analyzed. Chapter IV explains the methodology of how the software visualization tool was created. Specific topics included in this chapter are software requirements analysis, design, testing, and validation. Chapter V provides an analysis of the validation process with respect to the visualization techniques. Finally, Chapter VI makes conclusions and suggestions for future extensions of this research.

2. Literature Review

This chapter provides a background into the field of visualization and, more specifically, information visualization. Then, information visualization as an emerging technology is covered, followed by a section on human perception and pattern recognition. Next, several examples of organizations using visualization are presented. Also some general visualization guidelines along with applied visualization techniques are discussed. Finally, multivariate visualization techniques are presented. This research effort focuses on information visualization and the techniques used to display multidimensional data.

2.1 Introduction

Data is the organizational challenge of the next century (Teresko, 1996: 66). Teresko adds that "competitive success, even business survival, will depend on the ability to quickly turn it into comprehensible information" (Teresko, 1996: 66). Information technology has grown beyond the mere ability to just automate data processing. Part of its new potential is data visualization, the art and or science of accessing, structuring, and graphically presenting the information contained in large, complex data sets. The visualization process emphasizes a very deep, penetrating look into the actual structure of the data. Based on this more in-depth look, inferences drawn from the visualization are based on more suitable, detailed knowledge of the particular system being studied (Ellison, 1994: 622). Data visualization represents a special opportunity as organizations struggle to expand information technology structures to accommodate the future. Industry is

clearly seeing the need to meet the challenge of too much data and too little information. Ultimately, all data analyses lead to the creation of either figures or tabulated data, and a figure is usually better at conveying information obtained from an analysis than are tabulated data (Brown, 1994: 95A). The successful organizations will take further steps to help reduce the information gap by using data visualization to support these change initiatives. Visualization displays make larger data sets more understandable, not only at the end of an analysis, but also during it, which provides valuable insights into future inquires (Richman, 1996: 52).

2.2 Background

For the past thirty years computer graphics have enabled product designers and engineers to see their designs without having to build a physical prototype. The aim with this scientific, or data visualization, is to present numerical data in such a way that it is easily comprehensible and understood by the scientist, engineer, analyst, or user. The amount of data available is constantly increasing in line with improvements to computer processing and communication speeds (Kohn, 1994: 37). The data sets are becoming larger, more complicated, and dynamic. In order to gain a good idea into the trends in these data sets, it is much easier to have a graphical representation of the numbers, and preferably one that will change over time as new data becomes available (Kohn, 1994: 37).

Scientific visualization aims to devise algorithms and other associated methods that transform, through a variety of techniques, massive scientific data sets into visual displays such as pictures and other graphical representations, that facilitate comprehension and interpretation of the data set (Samtaney, 1994: 20). The ultimate goal of visualization,

claims Samtaney, is to help the user understand and be able to better analyze the data. Technology is expanding tremendously with respect to faster computers, more sophisticated sensing devices, and higher bandwidth communication channels. Therefore, data is being produced in greater amounts. The information obtained from the data must be presented to the decision maker in a form suitable for thorough understanding. There is an urgent need for better tools to automatically search for and compare the needs of the user with the most appropriate visual representation.

Engineers, scientists, and even physicians need an alternative to interpreting numbers and symbols (Kaufman, 1994: 18). Visualization provides that alternative by transforming simple numbers and symbols into a universally understood language of visual displays. The visual displays then enable both the researcher and practitioner to better observe and understand their data, computations, or models. Visualization helps in the process of extracting useful information from within the original, raw data set (Kaufman, 1994:18). Visualization encompasses a variety of strategies and techniques such as computer graphics, image processing, computer vision, computer-aided design, signal processing, user-interface studies, cognitive sciences, and computational geometry (Kaufman, 1994: 18). Kaufman further adds that "visualization technology continues to generate much excitement and enthusiasm and has already significantly enhanced the way scientists do science, the way engineers design, and the way physicians deliver health care" (Kaufman, 1994: 18).

The field of computer-based visualizations began more than two decades ago with the first reports of computer imaging being used to gain insights into data sets (Kaufman, 1994: 18). In the 1980s, important breakthroughs in the field of visualization led to the

birth of visualization as an actual discipline. In fact, the National Science Foundation has recognized the importance of visualization research and the need for the United States to maintain its leadership in this area (Kaufman, 1994: 18). Kaufman further adds that visualization is the subject of considerable research and development activity at all United States National Supercomputer Centers, at many academic institutions, at national laboratories, and within private companies throughout the world.

2.3 Visualization Concerns

Some specific concerns, when using visualization techniques to represent data, are the ability of images to represent information clearly, the dependency of visual and information perception on past memories, experiences, beliefs and culture, and the difficulty in making effective use of color (Gershon, 1996: 441). Images may have some disadvantages at times and therefore words are sometimes more effective or powerful than pictures. Gershon further states "to make full and correct use of what display, graphics, and visualization technologies can offer us, we need to take these considerations into account when generating images or when viewing them" (Gershon, 1996: 441). He further points out that for certain purposes, images do not need to portray reality exactly. However, when these situations occur, the viewers must be made aware of this fact, otherwise the pictures created might be worth "1/1000 of a word or even -1000 words." Gershon also mentions that not everything can be put into a visual form.

The educational systems focus on verbal literacy. However, with the advances in advertising, trick photography, and computer graphics, the need for visual literacy is on the rise (Fracchia, 1996: 442). Although this may not have been a critical issue in the past,

the potential for creating visual ambiguity is becoming a major concern. People become a high risk target of being fooled by what they see. Fracchia points out the need to understand how pictorial ambiguity arises. Before being able to distinguish images, the basic components of visual language must be discovered. At this point, exactly what these would be is unclear, especially with respect to computer images (Fracchia, 1996: 442).

Words and images speak to different parts of a person's unique experiences (Glassner, 1996: 442). Glassner believes these media are complimentary to one another. He states "words often fail to describe images, but images just as often fail to capture what can be said by words." Although a picture may be worth a thousand words, those words may not be the same from one person to the next (Mones-Hattal, 1996: 442).

2.4 An Emerging Technology

Recent developments of computer visual display hardware on one hand and computer graphics and visualization methods and software on the other have generated new interests in using images and visual representations to view data. People in the graphics and visualization community have taken the saying "one picture is worth a 1,000 words" and given it additional meaning by perceiving it to mean not only that images could portray anything that words can, but more so, that images could do it better than words (Gershon, 1996: 441). Gershon goes on to state the following with respect to images:

- Images are more powerful than words.
- One could pack more information into a given space using images than by using words to describe the same information.
- Images can convey information that words cannot.

- Images can also deliver information more quickly and efficiently than by using words.
- Words could be fuzzy. Images show the truth as it is.

Computer hardware and software are making the visualization of data increasingly more available. With these improvements also comes the capability of using more and better visual cues to reveal and display the underlying and often hidden meaning within the data. The field of visualization is an emerging technology for understanding large, complex, information rich data sets (Eick, 1996: 74). Data visualization has proven effective in deciphering many types of scientific and engineering data and thereby facilitating human comprehension of large, complex data sets (Heath, 1995: 21). Spreadsheets first revolutionized the ability to transform small amounts of data into understandable information whereas visualization is revolutionizing the way large data sets are understood. The ability to generate, collect, and store data has experienced a tremendous growth explosion over the last decade (Eick, 1996: 74). As a result, organizations are becoming more data-driven and data is being recognized as a very strategic asset. By visualizing data, a better understanding of the underlying principals hidden within the information can be seen. Because the data is better understood, the decision process is improved and hence better informed decisions can be made.

When numbers or symbols are translated into graphics or some other visual representation, massive amounts of data can be appreciated and understood. Even more importantly, relationships otherwise hidden within the raw data sets can be more easily revealed through visualization (Booker, 1993: 28). Visualization attempts to extend comprehension of the data set. This emerging technique will help unleash the full potential

of human and computer interaction. Ben Schneiderman, head of the Human-Computer Interaction Laboratory at the Center for Automation Research at the University of Maryland in College Park, Maryland believes the promise of visualization technology is enormous. He was quoted in Ellis Booker's article "A New Method of Seeing?" stating "Our capacity to perceive patterns is enormous, and I believe the computer industry has inadequately attended to that. We still have the legacy of the teletype and the fantasy of artificial intelligence" (Booker, 1993: 28). Making visualization systems work requires the collection of raw numerical data and an intuition about how the data interacts. Although the potential is great, visualization is still an emerging technology.

2.5 Human Perception

Pattern-seeking, which is a natural and very important part of the visual thought process, is the first step of a two-step process (McKim, 1980: 60). During the actual act of "seeing", this first step is to perceive an overall pattern void of any particular detail. The second step, which proceeds according to personal needs and interests, is to analyze the overall pattern for details. At first glance, the viewer attempts to roughly "draw" the overall relationships. Next, the viewer develops the precise details, seeking to bring into focus that which is perceived to be important.

2.5.1 Gestalt

Gestalt is a German word that has no exact equivalent in English (McKim, 1980: 60). Words such as form, shape, configuration, or pattern would be close matches. Toward the end of the nineteenth century, a group of Austrian and German psychologists performed research and formulated theories about the role of pattern-seeking in human

behavior. "Gestalt psychology" is a field that has evolved from these studies and has been productive with respect to visual perception. Gestalt psychologists believe that perception inherently behaves as an active force, as opposed to a passive response. According to this view, every perceptual image consists of more than the sum of its parts. Each image possesses a "gestalt" or a patterning force that holds the parts together. This becomes extremely relevant to this research because the natural sequence of all visual thinking processes is to seek patterns, then analyze.

2.5.2 Data Visualization

James Martin, founder of James Martin and Co. in Fairfax, Virginia, states "data visualization helps improve comprehension of complex data. Data visualization is a pictorial representation of data; it can be sliced, diced, or plotted to create a landscape of the data" (Martin, 1996: 37). He further adds that the goal of visualization is to transform large, complex data sets into patterns which the human mind is good at recognizing. The beauty of data visualization is that is can be exploited for a wide variety of functions. It can be also be used to create complex simulations and "what if" scenarios. Organizations that can recognize, capture, analyze, and use information quickly will have a competitive edge (Martin, 1996: 37). Those organizations using data visualization will be able to see trends forming and more importantly act appropriately on them. They will be able to take full advantage of opportunities and insights other organizations might not even see.

2.5.3 Computer Assistance

Numbers are the foundation for virtually all scientific fields; however, so many are being produced that they can easily overwhelm the researcher (VanDam, 1992: 2). Identifying patterns in such large, complex data sets can become a tedious and time consuming task. To help reduce this overwhelming task, the computer can be used. The computer, with appropriate visualization software, can digest vast amounts of data and transform them into sharp, graphic forms. Consequently, instead of having to scan through millions of numbers representing different variable values, the data can be easily viewed as a visual display of information representing the data set. This aids in the overall understanding of the data. Advocates of scientific visualization, the branch of computer graphics dedicated to conceptualizing research results, claim a correspondence between the human brain's highly developed visual abilities and the power of computer images that are capable of incorporating many variables (VanDam, 1992, 2).

2.5.3.1 Information Explosion

In this era of the information explosion, there exists the need to take advantage of the power that is provided by the human's visual processing system. Information visualization technology will help in understanding the content of the unlimited amount of information as well as improve upon the analysis of that information. Visual representation of the information has the potential to be a powerful force to aid improved understanding of the data (Rose, 1996: 442). In order for the visual representation to reach its full potential, the visual display must be based on a comprehensive and information-based structure.

When information overload occurs to the point that a user "cannot see the forest for the trees", visualization offers a way of identifying and analyzing the underlying patterns in the data. Data visualization is about comprehension, not building cute graphics. Visualization encompasses a variety of techniques that create the displays of abstract numerical data and statistics into a graphical form. Individuals comprehend information most intuitively through visual senses (Weber, 1993: 121). With half the neurons in the brain dedicated to visual processing, visualization images provide the greatest "bandwidth" to capture this. Therefore, by offering a visual picture of the data, along with its corresponding internal relationships, the process of visualization makes it easier to understand information that may well be too complex to perceive numerically. Weber further points out the benefits of visualization to include the ability to analyze numerical data graphically, view data multidimensionally, and comprehend complex data more easily.

2.5.3.2 Complex Systems

The complexity of today's systems has necessitated an increasing amount of computer assistance to ensure efficient, accurate, and reliable operations. Although the machines are able to process and deliver more data faster than ever before, the vast magnitude of numerical data can easily overwhelm an analyst or decision maker. A computer's capability to perform complex calculations and handle massive quantities of data continues to accelerate exponentially (Douglas, 1994: 19). However, the ability of humans to keep up with the flow of results becomes a very crucial issue that must be addressed. At the center of this problem is the basic underlying mismatch between the

way computers “think” and the way humans do. Computers are capable of quickly absorbing and comprehending massive amounts of information and fine details. Humans, however, tend to look for the “big picture” which can unfortunately be hidden within the details. The emerging field of scientific visualization is attempting to bridge this gap by creating visual displays that help humans comprehend in a more natural and intuitive way (Douglas, 1994: 19). Martin Wildberger, who oversees scientific visualization research in the Electric Power Research Institute’s (EPRI) Strategic Development Group believes if you want a human to understand something, arrange the data so that a pattern can be recognized; show relationships between variables; omit things that aren’t needed; and above all else do not overload the user (Douglas, 1994: 19).

2.6 Visualization Examples

Scientists use visualization to analyze the results in experiments in fields as far ranging as pollution studies, semiconductor physics, and drug design (Weber, 1993: 121). In engineering fields, visualization can provide very quick solutions in design, development, and production planning. Civic and voluntary organizations can use visualization in order to obtain better demographic type data. In business, the use of visualization techniques can help determine buying patterns, sales information, or investment returns. The examples provided here clearly demonstrate how visualization has been integrated into a variety of diverse applications and more importantly the valuable contribution they have provided.

2.6.1 Army Battlefield Visualization

On 3 May 1995, Lieutenant General Paul E. Menoher, Jr., (then Deputy Chief of Staff for Intelligence) briefed General Gordon R. Sullivan (then Chief of Staff of the Army), on the concept of a battlefield visualization. As a result of this briefing, General Sullivan directed that a battlefield management plan be developed and a battlefield visualization advanced concept technology demonstration be conducted (Menoher, 1996: 138-139). Subsequently, an advanced technology demonstration was approved by the Army Science and Technology Working Group. As soon as September 1995, XVIII Airborne Corps soldiers were able to portray enemy forces overlaid on a high resolution, three-dimensional, virtual display of a corps exercise area in Panama. Eventually, the model was enhanced by including friendly force data and modular, semiautomated forces. This enhancement created a more realistic planning, war-gaming, and rehearsal capability against opposing forces using simulation models. Battlefield visualization shows great potential for the future by providing commanders the ability to see their physical battlefield and the friendly and enemy forces on it. The commanders gain a better insight into the data, develop better planning strategies, and can "virtually" rehearse an operation. Battlefield visualization will help achieve this dominant awareness and the Army is moving aggressively to provide each Army warfighter a viable battlefield visualization capability (Menoher, 1996: 139).

2.6.2 Mathematical Visualization

Although the majority of mathematical literature has been predominantly algebraic for most of the twentieth century, the use of visualization has been growing tremendously.

Mathematical visualization is the emerging art of creating tangible displays using abstract mathematical objects and their transformations (Hanson, 1994: 79). This particular process has long been a mainstay of mathematical reasoning; however, the use of interactive computer graphics systems have opened a new generation in the visualization of pure geometry. Although this technology is rapidly advancing, it is easy to abuse visual displays to support arguments that would not on their own stand up to formal proofs. Also, certain problems involve properties of space too complex to actually display and must therefore still be treated algebraically. There are many mathematicians who feel that computer graphics technology will significantly influence the progress of mathematics in the future. Mathematical visualization is a rapidly emerging field of visualization science. Some of the same skills and techniques might be appropriate in areas such as interactive interface design, computer graphics, efficient algorithms, and data management.

David Hoffman, head of the Scientific Graphics Initiative at the Mathematical Science Research Institute in Berkeley, was among the first to explore the use of visualization in mathematics (Kohn, 1994: 37). Hoffman was trying to incorporate visualization to help him understand more about the behavior of minimal surfaces, objects that resemble soap films. Interestingly, he discovered a more useful aspect of his visualization technique, explaining complex ideas to colleagues from other departments. For example, Hoffman was collaborating with a materials scientist from the Massachusetts Institute of Technology (MIT) on a chemistry project. There was a huge communications barrier between the two scientists. They were from different communities and mathematical descriptions of the solution were too difficult to be understood by the

chemist. Visualization bridged the communications gap in understanding. Once they had the image, they didn't have to communicate through equations.

2.6.3 Environmental Protection Agency

Scientific visualization techniques from specific research efforts to major policy-setting and regulatory activities have become an important component of the United States Environmental Protection Agency's (EPA) High-Performance Computing and Communications Program (Rhyne, 1995: 94). This requires EPA regional and program offices, as well as state environmental protection agencies to be provided with desk top scientific visualization capabilities. The EPA's Scientific Visualization Center is currently developing and testing an application for the world wide web. The application will include computer-based instruction on applying scientific visualization techniques to EPA data sets into readable formats suitable for scientific visualization software, guidelines for preparing an animation, tools for integrating visualizations with geographic information systems and other relevant information pertaining to visualizing environmental science data sets. A number of scientific visualization researchers have been exploring the feasibility of building intelligence into the visualization software. In doing so, a user would specify an area of interest, describe the data parameter, and determine an analysis objective. The intelligent software tools would then suggest and describe different visual data representations. Once a user selects the desired representation, the software would then automatically create an appropriate visualization. With these visual display tools, environmental scientists, policy analysts, and decision makers will be able to collaborate

via the Internet to examine and control complex ecosystem problems that span multiple state boundaries and international borders.

2.6.4 Financial Visualization

In the financial community, technology has created a vast and complex world. As a result, visualization-based technology is growing rapidly and is projected to proliferate on Wall Street (Yrastorza, 1996: F9). Visualization software will improve the comprehension of large quantities of complex, financial information by providing the user with a visual representation of data using color, size, shape, dimensionality, and even motion. These particular representations enable users to more quickly see patterns or relationships within large volumes of data; whereas the conventional methods like spreadsheets, rows, and columns, cannot detect these patterns.

Although visualization technology was first used in scientific research, data visualization made its leap to the financial community around 1992. Since then, several international giants in banking and securities, including J. P. Morgan, Lehman Brothers, Chase Manhatten, Merril Lynch, and Citibank, have employed visualization technology (Yrastorza, 1996: F9). Yrastorza went on to quote Jeff Saltz, vice president and manager of the data visualization stripe at J. P. Morgan & Co. stating "visualization is a tool for allowing you to cut through masses of data to discover patterns not otherwise accessible to you. The result is higher quality information, which clearly relates to better decision making" (Yrastorza, 1996: F10). The return from using visualization in the financial community has been dramatic. For instance, J. P. Morgan's equities trading profits soared almost 117% in 1995 and Saltz stated visualization contributed to the gain.

The company further feels visualization technology has given them a competitive edge and has made a long term commitment to developing and extending visualization-driven applications. Nations Bank, Inc., a banking company in Charlotte, North Carolina, is using visualization software to identify likely buyers of future products and profitable buyers of current offerings (Richman, 1996: 52). The visualization tools are becoming more necessary as a response to the ever increasingly ambitious analyses sought by users and decision makers. Current query tools, which elicit textual or tabular responses, are no longer adequate for multidimensional, complex data sets.

2.6.5 Sun Microsystems

Data visualization techniques have begun to play an important role in a new corporate-wide quality initiative at Sun Microsystems, Inc. (Teresko, 1996: 67). The mission was to develop and implement a scaleable tool that could present a macro view of information to executive users and more detailed information to Sun's 15,000 employees in North and South America, the Pacific Rim, and Europe. Sun based its quality-information system on data visualization software from SAS Institute Inc., Cary, North Carolina. One of the most important features allows viewers to explore down into the information hierarchy for data on the consolidated corporate dissatisfiers and geographic or product specific information. The data visualization software also has the capability to relate what the dissatisfiers are costing Sun annually. The system has documented a reduction in the overall number of customer dissatisfiers, despite substantial growth in the volume of shipments and overall installed user base. In some instances, growth was as high as 20 percent, but the absolute number of dissatisfiers was reduced.

2.7 General Visualization Guidelines

There are several guidelines which should be considered when designing a visualization representation of some data set. The goal should be a graphical display that in the simplest terms shows the data. The graphic should in no way distort what the original information was intended to say. The display should serve a reasonably clear purpose and be closely integrated to the original data set. Graphical excellence is "that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space" (Tufte, 1983: 51). Tufte provides additional insights by discussing the importance of graphical integrity, the use of maximizing your data ink, and some important thoughts on the use of color. Lastly, visualization goals are identified with examples of their utility.

2.7.1 Graphical Integrity

Graphical integrity must be a major goal when attempting to represent a data set visually. Graphics should not distort the original data set. In doing so, the original intent is lost and the meaning to be conveyed goes unnoticed. The representation of numbers should be directly proportional to the numerical quantities being represented. Using clear, detailed, and thorough labeling, the visual representation can be applied to reduce the chance of graphical distortion and ambiguity. Care must be taken with higher dimensional data such that the number of information-carrying variables or dimensions depicted should not exceed the number of dimensions in the original data set. The graphical representation should never quote data out of context. Many graphic artists believe that statistics are boring and tedious; therefore, they believe that graphics should be decorated in order to

pep up or animate the data. However, too often this leads to exaggeration of the original data (Tufte, 1983: 79). Tufte provides a short checklist of questions to address with respect to data integrity (Tufte, 1997: 62):

- Is the display revealing the truth?
- Is the representation accurate?
- Is the data carefully documented?
- Do the methods of displays avoid deceptive readings of the data?
- Are appropriate comparisons and contexts shown?

2.7.2 Data Ink

Data graphics should draw the viewer's attention to the sense and substance of the data, not to something else. In order to accomplish this, a large proportion of the "ink" used in a visualization should present data information (Tufte, 1983: 93). Tufte defines the data ink ratio as the amount of actual data ink divided by the total ink used to print the graphic. The obvious goal here is to maximize the data ink ratio, within reason of course. Too often, decoration or non-data ink just contributes to clutter. When too much clutter appears, the graphical representation is diminished in value. The user may no longer see underlying meaning or patterns in the data. The data graphic should be able to stand on its own merits and not have to rely on decorative clutter. The graphic should be "user-friendly", allowing the viewer to easily understand and appreciate the intended purpose of the graphic.

2.7.3 Color

Color is a powerful aid to visual data representations, if used appropriately. The appropriate use of color is very important for effective visualization. However, the proper use of color is one of the most abused techniques used in data presentations. With respect to using color, Tufte states "above all, do no harm" (Tufte, 1991: 81). Putting good color in a good place is a complex process. Color can be used to label, measure, represent or even decorate. Color may also be very effective by omission, for example to distinguish a single element from a group or when comparing two components (Robertson, 1991: 338). Separating two data sets, or two phases of an interaction sequence, can be very effectively performed by depriving one of its color. Many of the large number of possibilities can degrade the representation rather than improve it (Levkowitz, 1991: 336). Because of this risk, the use of color has become very controversial. However, the increase in color graphics capabilities and in the demands put on visualization techniques have increased the interest in the use of color. Tufte mentions four rules when using color (Tufte, 1991: 82-90):

- **Rule One**: Pure, bright or very strong colors have loud, unbearable effects when they stand unrelieved over large adjacent areas. Used sparingly on or between dull background tones can have extraordinary effects.

- **Rule Two**: Placing of light, bright colors mixed with white next to each other usually produces unpleasant results.

- **Rule Three**: Large area backgrounds or base-color should do their work quietly, allowing the smaller bright areas to stand out vividly.

- **Rule Four**: If a picture is composed of two or more large, enclosed areas in different colors, the picture falls apart.

An important questions rises - what palette of color should be chosen to represent and illuminate information? There is not an immediate ordering scheme to colors. Colors found in nature, especially those on the lighter sides such as blues, yellow, and grays, are familiar to the eye and usually produce excellent results in graphical displays. Some combinations of bright colors do not work well and should be avoided (Brown, 1995: 130). For example, putting bright blue next to either bright green or bright yellow causes the edges to be blurry. Similarly, the use of bright red in combination with either bright green or bright blue causes an appearance of vibration and possible problems at the edges where the colors meet (Brown, 1995: 130).

As with the other techniques discussed, the use of color in a graphic can easily be misused. Relying on color difference as the sole method for sending a message and the arrangement of color fields that make different colors look alike can create confusing and disturbing graphics. Color can also deceive the viewer. The same color appears to be different if the color areas are of different sizes. Also, approximately eight percent of the male population and less than one percent of the female population suffer from some form of anomalous color vision (Meyer, 1991: 338). Color can also have different meaning depending on the viewer's perception. Brown displays associations of color by culture and profession using the following tables (Brown, 1995: 135):

Table 1. Some Cultural Associations of Colors

	Association to Colors			
Culture	Red	Blue	Green	Yellow
Japan	Anger Danger	Villainy	Future Youth Energy	Grace Nobility
United States	Danger	Masculinity	Safe	Cowardice Caution
France	Aristocracy		Criminality	
Egypt		Virtue Faith Truth	Fertility Strength	Happiness Prosperity

Table 2. Associations of Color by Professional Group

	Professional Group		
Color	Process Control Engineers	Financial Managers	Health Care Professionals
Blue	Cold Water	Corporate Reliable	Death
Turquoise	Steam	Cool Subdued	Oxygen deficient
Green	Nominal Safe	Profitable	Infected
Yellow	Caution	Important	Jaundiced
Red	Danger	Unprofitable	Healthy
Purple	Hot Radioactive	Wealthy	Cause for concern

2.7.4 Goals

A primary objective of visualization is to create complete images that speak to the viewer clearly. Although immediate understanding may not always be attainable, working in this direction helps reduce the chance that the image will communicate poorly, imprecisely, or not at all. Therefore, a visual image should facilitate communication of knowledge, not merely display or represent data (Keller, 1993: 12). Keller identifies seven categories of visualization goals and examples of their utility (Keller, 1993: 183):

- **Comparing**: images, positions, data sets, subsets of data.
- **Distinguishing**: importance, objects, activities, range of value.
- **Indicating directions**: orientation, order, direction of flow.
- **Locating**: position relative to axis, object, map.
- **Relating**: concepts, e.g., value and direction, position and shape, temperature and velocity, object type and value.
- **Representing values**: numeric value of data.
- **Revealing objects**: exposing, highlighting, bringing to the front, making visible, enhancing visibility.

2.8 Applied Visualization Techniques

The field of information visualization is experiencing a rapid growth. Although still a relatively new field of study, several techniques have been developed in order to create the best possible visual representation of a data set. The following are some actual visualization systems and the techniques they use to visualize data.

2.8.1 AT&T Visualization System

AT&T researchers have formulated some design guidelines for creating information-rich visualizations of data (Eick, 1996: 82 - 85). These guidelines are based on practical experience, are linked to perception, and have enabled AT&T researchers to create a wide variety of innovative and effective displays of data. The design of these systems is based on allowing a user to discover relationships in the data, thereby enabling them to take some type of action based on their discovery. Eick identifies the following as the design guidelines to follow and then lists some potential areas being considered (Eick, 1996: 82-85).

2.8.1.1 Task Oriented

Because the needs of each user are unique, the best visualizations should be task oriented and answer interesting questions. Building a successful visualization requires knowing the tasks and analysis requirements of users, and being able to incorporate their domain knowledge.

2.8.1.2 Domain-Specific Representation

The representation determines how the items in the data set are rendered on the computer display. Eick states "representations should take full advantage of perceptual cues such as size, position, color, depth, motion, and perhaps even sound" (Eick, 1996: 82).

2.8.1.3 Color

Eick adds that "color processing in the human vision system is an independent perceptual process, making color a natural choice for encoding information." Another advantage of color is that it shows details, allowing additional information to be layered onto the display. Color also engages the user, is visually pleasing, and increases the overall appeal of the system.

2.8.1.4 High Information Density

This concept employs using every pixel to display data, thereby maximizing the information content of the visualization. In cases where the entire data set can be displayed, this method can help minimize the frustration of navigating, panning, or

zooming a representation. However, information-dense displays can often become cluttered with too much detail.

2.8.1.5 Interactive Filters

This approach attempts to solve the problem of too much data, referred to as display clutter, by reducing the amount of information displayed. Humans have pattern recognition capabilities and can be very efficient at manipulating interactive controls to reduce visual clutter. When a display becomes too busy for "easy" interpretation, interactive filters are a natural interface to solve the problem.

2.8.1.6 Multiple Linked Views

A visual representation of data can be enhanced tremendously through the use of interaction and linked views. Additional insights into the data set can be provided by being able to select and filter data in one view and have instant propagation to the other views. Then, linking multiple views interactively will provide an integrated visualization. This integrated view is then far more powerful than the sum of the individual views.

2.8.1.7 Drill Down

As users uncover interesting patterns, they need the ability to access the actual data values. As a user touches any item in the display with the mouse, the data values associated with that item should be displayed.

2.8.1.8 Animation and Motion

Animation is effective on large, time-oriented data sets by using each frame to represent a single time period. Eick states that "human perception is fine-tuned for motion detection, making animation a natural technique for scanning large data sets" (Eick, 1996: 83).

2.8.1.9 Potential Areas

AT&T's Visualization Research Group is currently investigating five additional areas of great potential: three-dimensional representations, sound, geography, time-oriented views, and built-in pattern recognition.

The added dimensionality of using three dimensions, when compared to two dimensions, is a natural way to pack more information onto the screen without overloading it. Perceptually, it is possible for three-dimensional displays to increase the information beyond what is possible with two-dimensional displays. This perception is accomplished by enabling the user's mind to create "virtual pixels."

Sound is another potential, but under-exploited medium for representing data. Sound has many aspects, such as pitch, timbre, and loudness. These aspects enable sound to encode data in many ways. Sound is fundamentally different from a visual display in four ways: it arrives through an independent channel; its bandwidth is lower than that of vision; it is immediate instead of persistent (as with visual representations); and it is serial in time (Eick, 1996: 83). Sound is especially suited for alerting, such as monitoring tasks. In animation, researchers have found that using voice to announce the passage of time is effective because it enables users to concentrate on the data display without having to

monitor the current time. In the business world, corporate data sets may have a geographic component, such as demographic information.

Future research by Bell Laboratories in developing spatial views with geographic information appears to show great potential. When data sets have a tendency to evolve through time, it has been considered advantageous to tailor the views in relation to time. AT&T researchers have already developed several time-oriented two-dimensional and three-dimensional views and is looking into using more sophisticated methods to take advantage of this naturally occurring phenomena.

Lastly, AT&T researchers are also trying to build automated tools to allow a user who identifies a particular pattern within the data set to detect all other occurrences of the pattern automatically and then to provide this information to other programs for action.

2.8.2 Visual Display of Parallel Performance Data

Data visualization is also being used to evaluate the performance of parallel computers (Heath, 1995: 21). The primary reason for using parallel computer systems is their high performance potential. However, being able to evaluate that potential is extremely difficult to realize. Users often must analyze and adjust parallel program performance. The complexity and volume of the data makes it increasingly difficult to interpret these systems. Hence, performance tools are needed to help alleviate the gap between raw performance data and significant performance improvements. Data visualization has proved effective in explaining many types of scientific and engineering data and facilitating human comprehension of large, complex data sets. In fact, some of the most successful parallel performance tools are based on visualization techniques. A

high-level abstract model lets visualization designers create displays in an integrated environment. The model then directly links these displays to parallel performance models. Heath identifies six important principles used in the development of the performance tool.

2.8.2.1 Context

To present information clearly, there must exist some context to which users can relate to the information. The perspective is the point of view from which information is presented. Typical perspectives might include the hardware, the operating system, and the application program itself.

2.8.2.2 Scaling

Scaling is extremely important when dealing with any type of scientific visualization. Scaling graphical views, as data sets become large, is a major visualization challenge. In order to handle this particular type of problem, several techniques have been developed. First, multidimensional and multivariate representation includes data with many attributes per data point. Although a multivariate representation is conceptually compact, the difficulty arises in the technical challenge for visual representation of multiple dimensions onto a flat two-dimensional surface such as a video screen (specific techniques will be addressed later in the thesis). Secondly, macroscopic and microscopic views are the precise level of detail that is represented by a given display. A macroscopic view deals with showing the "big picture", whereas a microscopic view displays much finer detail in the visualization. The third technique is macro/micro composition and reading. This particular display allows perception of both local detail and the overall global representation. With this visual display technique, fine details are discernible;

however, the details accumulate into larger, recognizable features which can convey meaning themselves. The fourth concept is adaptive graphical display which allows the adjustment of a display's graphical representation in response to a particular data set size. The main idea is the ability to reveal as much detail as possible without having visual complexity interfere with the perception of that detail. Display manipulation, the fifth technique used in scaling, is the ability to interactively modify a given display, through a series of techniques such as scrolling or zooming, in order to handle large amounts of data based upon the level of detail required for viewing. The last technique for scaling is called composite view. This is a combination of two or more views into a single view that is intended to enhance visual relationships among the views and present a better understanding of the underlying structure.

2.8.2.3 User Perception

There must exist some feedback between the user and the visualization tool. Important concepts in this category include: perception and cognition, observing patterns, and user interaction. Human visual perception can grasp patterns, distinguish variations, and classify objects through size, shape, color, and motion. The performance tool used color to provide an overall observation indicating processor utilization that can be easily perceived (Heath, 1995: 21). Using distinguishable patterns to portray information or relationships within the data set can be easily interpreted by a user. Lastly, the user should be allowed to make decisions regarding alternate views, levels of detail, and display parameters. This interaction enables users to customize the visualization for a given situation to increase their own understanding.

2.8.2.4 Comparison

Comparisons and cross-relationships between related views or representations can provide much insight into possible behavioral characteristics and more importantly their causes. Graphical techniques used are multiple views, small multiples, and cross-execution views. Multiple views are visual presentations of data using multiple displays from different perspectives. Sometimes, a single visualization or perspective may only display a portion of the actual behavior and possibly even distort the analysis of the data set. Creating the ability to view the same data set from a variety of perspectives can provide a more well-rounded, better understood impression of the data set, yielding more useful insights. Small multiples are a series of images showing the same combination of variables indexed by changes in another variable. One particular example of this technique is animation, where the indexing is over time. Finally, cross-execution views are visual comparisons of information, displaying differences with the different views.

2.8.2.5 Extraction of Information

Several techniques enable extraction of useful information from an abundance of data. Reduction and filtering consists of representing the data visually. This technique supports reduction by being able to show general trends rather than detailed numerical behavior. Clustering is a multivariate, statistical analysis and presentation technique for grouping or categorizing related data points. The focus is to classify data points or identify outliers in a multidimensional data set. The concept of encoding and abstracting is accomplished by using graphical attributes such as color, shape, size, and spatial orientation for displaying additional dimensions. Such overloading can be easily abused,

but, when used appropriately, can increase a flat screen's dimensionality. Lastly, separating information by using color, highlighting, foreground, background, and so forth, can identify visual differences among layers of information.

2.9 Multivariate Visualization Tools and Techniques

"The fascination with dimensionality predates Aristotle and Ptolemy who argued that space had only three dimensions. By the nineteenth century, the new mathematics of Riemann, Lobachevsky and Gauss unshackled the imagination and higher-dimensional geometries came into their own" (Inselberg, 1995: 405). Currently, most of the attention in visualization research has focused on data found in physical phenomena, which is generally limited to three or four dimensions. However, many sources of data do not share this dimensionality restriction. The critical problem which arises during analyses of multivariate data is providing researchers with tools which can be used to gain insights into characteristics of the data and possible previously unnoticed patterns. The following sections discuss tools and approaches currently being used to visually represent multivariate data.

2.9.1 XmdvTool

One particular system is called Xmdvtool which integrates several of the most common methods for projecting multivariate data onto a two-dimensional screen (Ward, 1994: 326). This integration of methods allows users to explore their data in a variety of formats with relative ease. Ward defines multivariate data as a set of entities E, where the i^{th} element e_i consists of a vector with n observations (x_{i1}, x_{i2}, ..., x_{in}). Each observation (variable) may be independent of or interdependent with one or more of the

other observations. The variables may be discrete or continuous in nature, or take on symbolic (nominal) values. Variables may also have a scale associated with them. When discussing the Xmdvtool system, Ward points out three multidimensional data visualization methods: scatterplots, glyphs, and parallel poordinates.

2.9.1.1 Scatterplots

Scatterplots are one of the oldest and most commonly used methods to project high dimensional data to a two-dimensional surface. In this method, N*(N-1)/2 pairwise parallel projections are generated, each giving the viewer a general impression regarding relationships within the data between pairs or dimensions. In Brand Fortner's book The Data Handbook an example of the scatterplot is provided using a six dimensional data set. Every combination of one dimension versus another dimension is plotted. The result is a 6 X 6 or 36 two-dimensional scatterplots (Fortner, 1995: 107). Figure 1 shows an example of this type of plot by holding the variable along the x axis constant and using the y axis to form each possible combination.

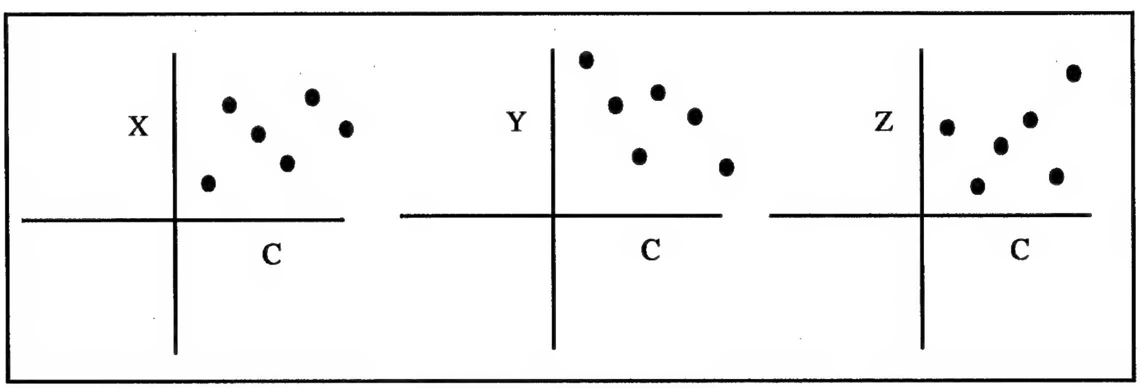


Figure 1. Scatter Plot

Many variations on the scatterplot have been developed to increase the information content of the image as well as to increase the ability to increase data exploration. The Xmdvtool generates the scatterplot by dividing the display window into an N by N grid, and each data point results in N^2 points being drawn, using only two dimensions per view. Columns and rows in the grid are labeled according to the dimension they report. One major limitation of scatterplots is that they are most effective when dealing with a small number of dimensions. As the dimensionality increases, the screen space available for each projection decreases. A couple of techniques to help alleviate this is using three dimensions per plot or providing panning or zooming capabilities. Other limitations include being restricted to orthogonal views and difficulties in discovering relationships which span more than two dimensions. However, scatterplots do include ease of interpretation and relative insensitivity to the size of the data set.

2.9.1.2 Glyphs

The second technique Ward discusses is glyphs. The definition of a glyph covers a large number of techniques which map data values to various geometric and color attributes of graphical primitives or symbols (Ward, 1994: 327). Some of the many glyph representations proposed over the years include the following:

- **Faces**, where attributes such as location, shape, and size of features such as eyes, mouth, and ears are controlled by different data dimensions.
- **Andrews glyphs**, which map data to functions (e.g. trigonometric) of N variables.
- **Stars or circle diagrams**, where each glyph consists of N lines emanating from a point at uniformly separated angles with lengths determined by the values of each dimension, with the end points connected to form a polygon.

- **Stick figure icons**, where the length, orientation, and color of N elements of a stick figure are controlled by the dimensional values.

- **Shape coding**, where each data point is represented by a rectangle which has been decomposed into N cells and the dimensional value controls the color of each cell.

In Xmdvtool, the star glyph pattern is used. Glyph techniques are generally limited in the number of data elements which can be displayed simultaneously, as each may require a significant amount of screen space to be viewed. Glyphs are an effective approach for distinguishing and comparing several variables at once (Ribarsky, 1993: 134). Glyphs are a powerful technique because they are able to rely on the eye's ability to perceive small differences in shape or position and to discern properties such as color and shape. The end result produces the capability to have several attributes that can be associated to variables while still being able to see each of the attributes separately.

2.9.1.3 Parallel Coordinates

The third multivariate technique presented by Ward is parallel coordinates. Parallel coordinates is a technique originated in the 1970s which has been applied to a diverse set of multidimensional problems. In this particular method, each dimension corresponds to an axis, and the N axes are organized as uniformly spaced vertical lines. A data element in N-dimensional space projects itself as a connected set of points, one on each axis. Points lying on a common line or plane create readily perceived structures in the image. The major limitation of the parallel coordinates technique is that large data sets can cause difficulty in interpretation. As each point generates a line, lots of points can lead to rapid clutter. Also, relationships between adjacent dimensions are easier to perceive than between non-adjacent dimensions. The number of dimensions which can be

visualized is fairly large, limited by the horizontal resolution of the screen. However, as the axes get closer to each other, it becomes more difficult to perceive structure or clusters.

2.9.2 Alignment Viewer

Information visualization continues to face challenges by the need to represent abstract data and the relationships within the data (Chi, 1996: 133-139). Information visualization is becoming even more important as researchers discover different domains of multidimensional data. Various techniques have been developed to map multidimensional data to three-dimensional scenes, especially since data from different fields often requires dramatically different visualization techniques. An additional problem is when the data set is not just multidimensional, but also large, making screen clutter almost inevitable.

When dealing with large multidimensional data sets, additional techniques are needed to reduce the amount of screen clutter. If too much clutter is displayed, the effectiveness of visually representing the data is quickly diminished. Clutter can easily distort the view and the visual representation's intended meaning becomes unclear. The next section presents a description of a system called Alignment Viewer which was developed to visualize multidimensional data. After the system has been described, the techniques used to display the data and the techniques used to reduce the amount of screen clutter are discussed.

2.9.2.1 Description

Chi and his team of researchers presented a system for visualizing similarities between a single DNA sequence and a large database of other DNA sequences. Their initial efforts produced a system called Alignment Viewer (AV) which greatly improved biologists' ability to discover features in biological sequence similarity information by using a new technique for visualizing the data. Sequence similarity analysis is the comparison of a single sequence against known sequences kept in databases. Often, similarity reports include hundreds of thousands of alignments (matching segments between the input sequence and one of the database sequences). An entire report can be hundreds or thousands of pages long. Since the data is multidimensional and the graphical system is three-dimensional, the researchers had to develop a creative way to depict the multivariate data. Seeing the potential of information visualization, the biologists on the research team sought additional ways to visualize other information in the data.

Responding to the need, the researchers developed a technique that allows the user to arbitrarily map any of the twelve variables onto the X, Y, Z, and time axes. The researchers showed how biologically significant features can be investigated using the additional time axis. Moreover, the researchers showed how the users of the system can develop simple queries on the data set using visual query filters, and more importantly how the filtering techniques reduced the clutter in the information space. Case studies were supplied demonstrating the power of the combined technique in finding, extracting, exploring, and analyzing features within the data that were previously difficult to find. The researchers further showed how the added time axis and visual query filters provided

an effective way to analyze a particular data sequence. By extending Alignment Viewer with a set of powerful visualization techniques, based on feedback from the biologists, the researchers enabled molecular biologists to more effectively study the intricate relationships between similar sequences.

2.9.2.2 Techniques Applied

To help reduce this inherent screen clutter, filters were applied on each of the variables to help clarify the analysis. A simple query based on a range for each variable helped reduce the overall clutter displayed on the screen. In the visualization community of research, a number of techniques have been used for displaying highly dimensional data. The techniques used by Chi and his research team included glyphs, worlds-within-worlds, and parallel coordinates. With respect to glyphs, two or three variables corresponding to a particular piece of information are used to position an icon or marker representing the data, while a number of other variables are encoded by the marker's size, color, shape, etc. (Chi, 1996: 134). Worlds-within-worlds describes when a point in three dimensions is first specified, then a second smaller frame is displayed at this point. A surface can then be drawn using a new coordinate system within the second frame. Parallel coordinates lay out major axes in parallel with each point represented by a line connecting each axis.

2.9.3 Glyphmaker

Another visualization and analysis tool is called Glyphmaker, which allows nonexpert users to customize their own graphical representations using a simple glyph editor and a point-and-click binding mechanism (Ribarsky, 1994: 57 - 64). Glyphmaker's general approach to data visualization and analysis is to let users build their own

customized representations of multivariate data and then provide interactive tools for exploring patterns and relationships within the data set. In particular, users can create and then later alter bindings to visual representations, bring in new data or glyphs with associated bindings, change ranges for data, and do these operations interactively. Users can focus on data down to any specified level of detail. Glyphmaker empowers its users by letting them use their own domain knowledge to create customized visual representations for exploration and analysis of the data.

Glyphmaker was built on top of Iris Explorer, the Silicon Graphics Inc. (SGI) dataflow visualization system. Glyphmaker was initially intended for users who do not fully understand their data and, in particular, do not know which visualization will best describe their data. Glyphmaker allows users to explore and then analyze their data. The power of using glyphs for visual representation of a data set is found in their ability to represent all types of multivariate data, especially those with variables that have significance dependence on two or three spatial dimensions. The overall goal when developing Glyphmaker was to create a general approach for building customized representations of multivariate data and provide interactive tools for exploring patterns and relationships within these data sets.

2.9.4 Information Visualizer

Information Visualizer is a system which was developed using information visualization techniques to retrieve, store, manipulate, and understand large amounts of hierarchical information (Robertson, 1993: 65). The system was designed using cone trees which are hierarchies laid out uniformly in three dimensions. Nodes are drawn

similar to three by five inch index cards. The top of the hierarchy is placed near the ceiling of the room and also serves as the apex of a cone with its children nodes placed evenly spaced along its base. The next layer of nodes is drawn below the first layer with their children represented in the cones constructed. The aspect ratio of the tree is fixed to fit the overall size of the containing "room." Each layer within the overall data structure has cones of the same height. This height is calculated by dividing the room height by the tree depth. The body of each cone is shaded transparently, in order that the cone can be easily perceived while not blocking the view of cones behind it. Robertson mentions that displaying text within the nodes was extremely difficult because the node text did not fit the aspect ratio of the "cards" very well. Therefore, text was shown only for a particular path selected. An alternate layout was displayed horizontally.

The hierarchy was presented in three dimensions to maximize effective use of available screen space and enable visualization of the entire structure. A two-dimensional layout of the same structure using conventional graph layout algorithms would not fit on the screen. If done in this manner, a user would need to either scroll through the layout or use a size-reduced image of the structure. Cone trees have been used in a variety of applications. For instance, they have been used for a file browser, organizational structure browser, and have been used to visualize a company's operating plan. Other potential applications include software module management, document management, object-oriented class browsers, and local area network browsers.

2.9.5 Kiviat Diagram

It is a much more difficult challenge to visualize the interaction of multiple variables. Computers can easily deal with numerous independent variables by assigning each one its own dimension in an abstract space where well-understood mathematical roles are used to calculate interactions. For most people, however, trying to visualize dimensions higher than three is almost impossible. There has been some success in visualizing multiple dimensions through the use of Kiviat diagrams or star plots (Douglas, 1994: 19). In these particular diagrams, the axis for each variable radiates from a central point, and the plotted values along the axes are connected by lines to form an enclosed figure. For systems with numerous critical parameters, however, Kiviat diagrams take such complicated shapes that characteristic patterns which might be indicating some specific problems are indistinguishable. The diagrams are attempting to reduce the complexity by "projecting" multidimensional data onto a two-dimensional plane. However, this reduction technique may actually lose information that a user might find very useful. Figure 2 shows two Kiviat diagrams: A is a simple geometric shape created by plotting normalized values for the assigned variables on intersecting axes; B is a distortion of the regular shape and indicates a problem.

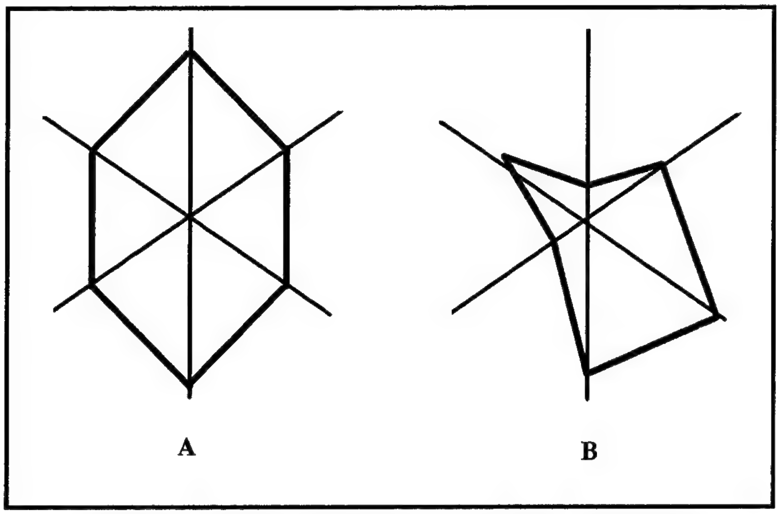


Figure 2. Kiviat Diagram for a Few Variables

2.9.6 Visualization of N-Dimensional Geometry

Understanding the underlying geometry of a multivariate problem can provide crucial insights into what is possible and what is not (Inselberg, 1995: 408). Another method to visualize higher dimensions in such a way that provides important opportunities for human pattern recognition, while retaining all the important mathematical information contained in the original multidimensional space, has been developed by Alfred Inselberg and his colleagues at the IBM Scientific Center in Santa Monica, California (Douglas, 1994: 20). The method is founded on the concept of parallel coordinates. Parallel coordinates, as previously mentioned, are vertical axes evenly spaced along a horizontal line, one axis for each variable. As the values are plotted on each axis and connected by lines, a point in an n-dimensional state space is viewed as a broken line across the n parallel axes. One easily grasped aspect of Inselberg's visualization scheme is the natural way it displays limits. If parallel axes are assigned to key system parameters, then each operational state of the system is represented by a line connecting the parameters' current values. Both upper and lower limits can be set for each parameter. Once the limits have

been established, the points can be connected, forming an envelope for the line showing current operations. A shift in this line toward operating limits along certain axes may then be interpreted as indicating particular problems within the data set. Figure 3 illustrates the use of using parallel axes for many variables. In this example, each axis represents a bus on the transmission network, and plotted lines connect values of power flow. Abnormal conditions are signaled when actual values extend beyond the thermal limit envelope indicated by lines at the top and bottom of the graph (Douglas, 1994: 21).

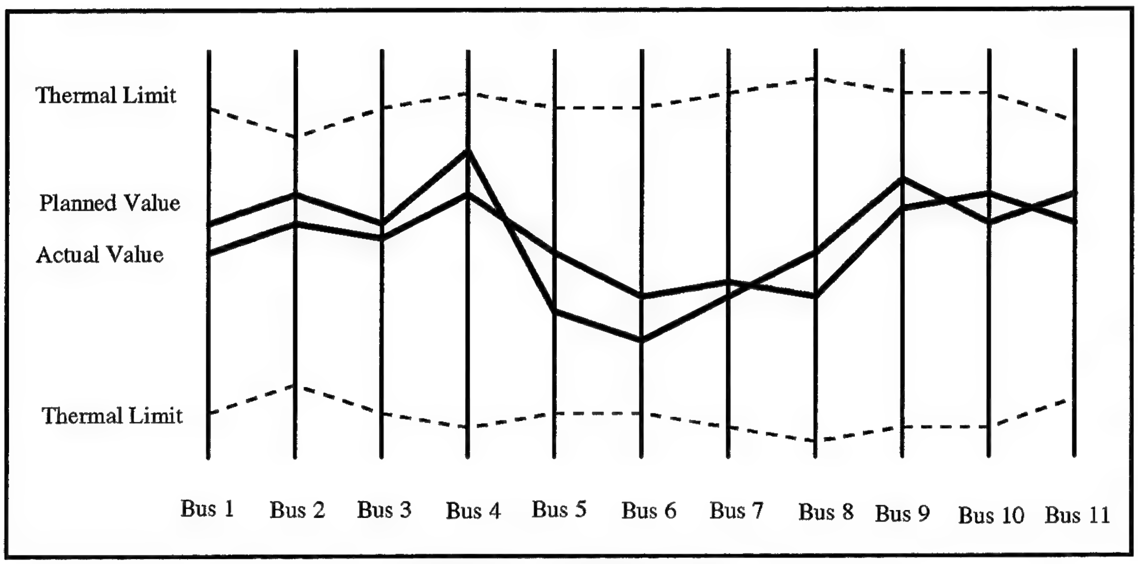


Figure 3. Parallel Axes

2.9.7 Iconograph Approach

The iconographic approach provides a general structure for the specification and creation of integrated displays of multiparameter images (Levkowitz, 1991: 165). The iconographic approach has been applied successfully in several areas such as medical imaging and computational fluid dynamics (Grinstein, 1995: 406). This particular approach appears similar to that of using glyphs. The approach extends the concept of

mapping data to a pixel to that of mapping data to an object, or icon, whose attributes such as color, geometry, reflectivity, opacity, and sound, are under the control of the various fields of the data itself. This particular technique allows the information or represented object to have an increased dimensionality. Variations of this icon have been used to display other forms of application information. This icon, in its many forms, enhances the human's ability to perceive line orientation "pre-attentively." Pre-attentive processing of visual elements is the ability to sense differences in shapes or patterns without having to focus attention on the specific characteristics that make them different. The greatest advantage of these iconographic displays is their ability to translate possible nonvisible and undiscovered statistical structures in the data into potentially evident visual structures. Humans have the capability to discriminate textures very effectively and use variations in texture as important sources of information. Hence, the observer can detect and recognize objects, and see statistical properties in the data as qualities of the displayed texture.

2.9.8 Grand Tour Methodology

The grand tour is an underutilized methodology for visualization of multivariate data (Asimov, 1995: 407). The methodology is based upon projecting the data to a two-dimensional subspace and displaying it onto the computer screen. The process is repeated by picking another nearby two-dimensional subspace, and then another, etc.. The sequence of subspaces should be chosen so that eventually it passes arbitrarily near to every possible subspace. The result of this visual display is an animation of two-

dimensional projections of the original data. By watching this animation, an analyst can often find patterns in the data that would have been undiscovered by other methods.

2.9.9 Geometry Approach

Visualization implies creating a pictorial form for the data. The geometry composing such a display, as with the data itself, can be categorized by its dimensionality (Treinish, 1993: 132). In order to appropriately visualize the data, a complete and thorough understanding of the data's characteristics is essential. Data is composed as a function of independent variables or, in other terms, its dimensionality. Some of the more complex data sets can consist of several dimensions. Treinish defines a parameter as the data or function itself or dependent variables, consisting of one or more values or rank. Rank zero (0) is defined as a scalar; it has one value. Rank one (1) is a vector which has three values in three dimensions. Even further, more complex data sets could be classified as rank two (2) or higher. The following table list the dimensions of typical geometries used to compose visualization images (Treinish, 1993: 132):

Table 3. Dimensionality of Visualization Geometry

Dimensions of typical geometries used to compose images	
Dimension	Geometry
0	Point
1	Line
2	Polygon
3	Volume

Often, there will exist some type of relationship or association between the dimensionality of the data and its geometry. This corresponding geometry is called a mesh or grid. Furthermore, the association between the data's dimensionality and its geometry

describes the relationship of the size, shape, and organization of the data to the physical coordinate structure. These mesh structures show relationships between the dimensionality of the data and its geometry. Mesh structures can take on several different forms. For instance, a regular grid with evenly spaced grid coordinates could be used as a temperature map. An irregular grid might have gaps in coverage such as one displaying several satellite images' coverage. Ungridded mesh structures are just a series of scattered points. Lastly, a deformed grid is curvilinear and could be used to display pressure on an airframe. The data itself could have one or more variables dependent on the dimensions. Figure 4 shows some examples of mesh structures portraying relationships between the dimensionality of data and its geometry: a) a regular grid, such as a temperature map; b) an irregular grid, such as several satellite images with gaps in coverage; c) ungridded (scattered points), such as sales figures or rainfall in specific towns; and d) a deformed (curvilinear) grid, such as the pressure on an airframe (Treinish, 1993: 132).

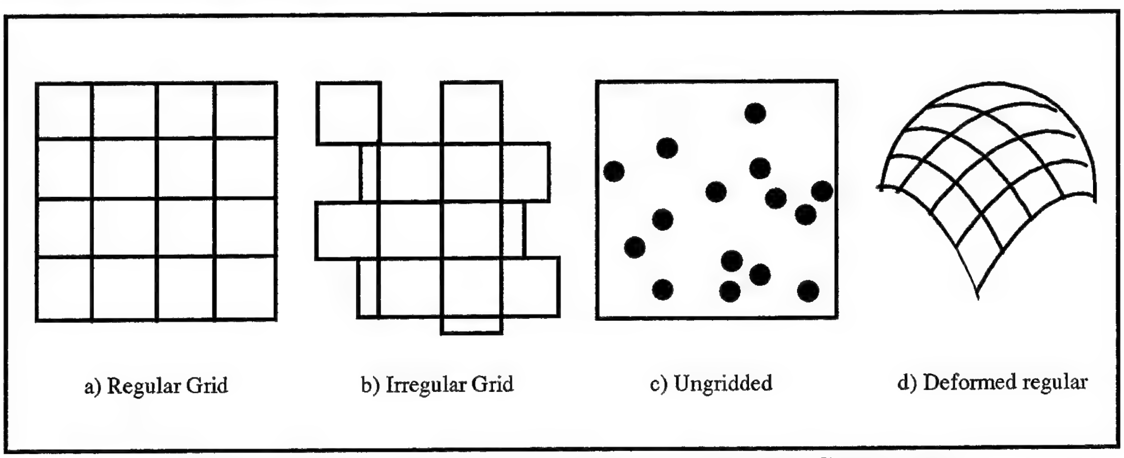


Figure 4. Examples of Two-Dimensional Mesh Structures

Treinish also offers a number of visualization techniques for representing multidimensional data (Treinish, 1993: 132 - 133). These techniques can be combined to represent multiple

parameters. It is even possible to embed a lower dimension in a higher dimension, such as inserting a line, plane, or surface in a volume. The dimensionality of a visual representation can be different from that of the data.

Table 4. Dimensionality of Sample Data and Visualization Techniques

Dimension/rank	Example of Data	Discrete Technique	Continuous Technique
0/0	Sales	Histogram	
1/0	Sales history	2-D scatter plot	2-D line plot
2/0	Ground temperature Medical image	Multiple x-y plots 3-D scatter plot Colored 2-D scatter plot	Isocontours
2/1	Ocean surface currents	Arrows	Streamlines
3/0	Atmospheric temp Stacked medical images	Colored 3-D scatter plot Isosurfaces	Volume rendering
3/1	Atmospheric wind	Arrows	Streamlines

2.10 General Design Guidelines

Whether the purpose of the visualization is for analysis or communication, the better the mapping between the data and visual form, which reflects the capabilities of the human visual system, the better the chance that the information will be detected and understood.

2.10.1 Introduction

To make certain aspects of the data explicit, it is necessary to know how information is encoded in a visualization, as well as the implications of the choices made in representing the data. The visual system should be able to decode information in several dimensions. By designing visualizations with human strengths and weaknesses in mind, it

is possible to exploit the visual system's ability to recognize structure and patterns and circumvent limitations in memory and attention. The fields of perception, psychophysics, and cognitive psychology have much to offer towards understanding the subtle workings of the eye-brain system (Marchak, 1993: 351). There are no hard and fast rules for applying these results to the visualization problem, since there are often complex interactions among the representation methods depending on the task. A graphical method for visualizing data consists of two parts: a selection of quantitative information to be displayed, and a selection of a visual display method to encode the information (Cleveland, 1993: 351). Some display methods lead to efficient, accurate graphical perception, while other display methods may lead to inefficient, inaccurate decoding.

2.10.2 Selecting a Technique

Selecting a technique for visualizing data and a medium for conveying the resulting image are only part of the process of communicating the meaning of data. The various components that depict the data must be selected, arranged, and displayed so that the intended meaning will be easily and correctly comprehended. A methodology or approach is needed for selecting visualization techniques; however, the discipline of information visualization does not yet have set formulas for selecting appropriate techniques. In describing a methodology, several key points must be addressed (Keller, 1993: 6 - 7).

First, the visualization goal should be identified. The meaning to be conveyed in the data must be known before a visual image is constructed. By knowing the desired goal, additional new sources of techniques may become apparent. Meanwhile, there is

now focus for determining if a prospective technique is likely to reveal the sought after meaning.

Secondly, all mental roadblocks should be removed. The data should be regarded as nothing more than numbers representing the information to be visualized. By associating the data as belonging to some application or having some structure beforehand, some possible techniques will never be considered.

Finally, the data must be well understood, distinguishing whether the information is data representation or phenomena (contextual-cue techniques). Data representation shows the data values independent of the phenomenon; the viewer must deduce the relationship to the phenomenon. Contextual-cue techniques relate the data values to the phenomenon being studied and add meaning to the visualization. Distinguishing between the two refines the visualization goal. The relationship between the visualization goal and the choice of techniques for accomplishing that goal is just beginning to be understood.

2.10.3 The Challenge

The challenge is learning how to efficiently and effectively use visualization resources and tools to reveal the true meaning within a given set of data. The following is a list of rules to consider with respect to a graphics or visualization package (Brown, 1995: 74):

- The software must be easy to learn and use, as defined by the targeted user's preference or ability.
- It must be easy to get data into and out of the system.
- The product should provide a complete set of general features and functions, all tightly coupled into a seamlessly integrated environment.

- The product should be "open," allowing for easy extensibility and customization.

- The software must work in a cross-platform environment, both PC running Windows and Unix workstations.

- The software should support both "what-if" (interactivity, point&click, direct manipulation) and "presentation quality" (scaleable, hardcopy, annotation, legends, axes) graphics techniques.

2.10.4 Overall Design

Basic visualization software has four types of tools or routines that are combined to produce an image (Wayner, 1993: 137). The software consists of tools for loading the data, transforming the raw data, rendering, and finally controlling the data.

- **Loading tools** - acquire the data by accessing the data from some external file and constructing a suitable internal structure to hold the data.

- **Transformation tools** - perform operations on data to modify the data in some way. Depending on the original raw data, these transformations can be simple or very complex.

- **Rendering tools** - use very sophisticated rendering algorithms to actually convert information into the actual pictures to be displayed.

- **Control tools** - enable the user to modify the actions of the other tools.

These four types of tools handle the basic tasks that visualization software must carry out. Good visualization software provides simple methods that work together efficiently.

2.10.5 Validation

Validation is concerned with answering the questions "does the visualization accurately, and effectively, represent the data?" and "does it help with insights and comprehension?". People who work in the fields of scientific visualization agree that it is possible to produce good, useful visualizations, and to produce very bad, misleading visualizations (Rushmeier, 1995: 422). There is also general agreement that visualization systems have characteristics that make them very useful for certain problems, and characteristics that make them essentially unsuitable for other problems. To date, very little work has been done to rigorously define what a good visualization or visualization system is (Rushmeier, 1995: 422). Currently, there are measures to guide users in generating reliable, accurate, and effective visualizations. Developers of visualization systems have no community-accepted standards and benchmarks to use in designing and validating their products.

Most visualization programmers come from the computer graphics community. This community generally values pretty pictures, which are not necessarily correct or informative. In many cases, visualizations are accepted if they look more or less right (Globus, 1994: 417). If the graphical image appears to look acceptable, the viewer assumes it is accurately portraying the original data. Usually, a user is called in to glance at the visualization and then make a few comments. The goal is that the visualization will increase the user's understanding. This can only be proven by experiments with human subjects. Such experiments are difficult to design and require collaboration with psychologists and/or human factors experts (Uselton and others, 1994: 417). Despite the

variety of visualization techniques available, there is little evidence showing the perceptual effectiveness or efficiency of different representation methods, such as which definable and recognizable visual representations are most useful for conveying specific information (Marchak, 1993: 352). Although modern visualization systems offer a wide range of visual dimensions, they typically offer no guidance on how these different visual dimensions are perceived (Rogowitz, 1993: 352).

2.11 Summary

Although the need to analyze large data sets of high dimensionality has been addressed primarily by scientists and engineers, it is a need that is now being shared by a wide range of non-scientists in business, government, and academia (Mihalisin, 1991: 171). The requirement to visualize and analyze data as graphics rather than text numbers is becoming the obvious choice as data sets continue to grow and become more complex. A satisfactory method of graphically presenting data involving more than three variables is not obvious and continues to be a major field of study.

Creating the kind of image that can provide the user with the most information from the data is the challenge. All of the many techniques used to analyze and comprehend data sets by providing visual representations are included in the broad definition of visualization. Interactive visualizations have become the trend as they provide tremendous benefits by interacting directly with data sets. Interactive visualization systems are more effective when the results are in some type of a dynamic visual form (Kaufman, 1994: 18).

3. Visualization Methodology

This chapter first introduces the challenges of information visualization. Next, a description of a currently used operations research analysis software application and its subsequent shortcomings of displaying multidimensional data is presented. The software visualization tool is discussed to include a description of each display created. An experimental procedure is developed to validate the design. The chapter concludes with sections covering how the validation data was collected and subsequently analyzed.

3.1 Introduction

To understand the challenge of visualization, a comparison between art and scientific visualization demonstrates why visualization can be so difficult. Both have the goal of communicating visually and symbolically. The artist, using time established tools such as canvas, brushes, and oils to illustrate a point of view, benefits from centuries of knowledge. However, the tools of scientific visualization, as well as the knowledge to use them, are still evolving. The computer's power is now needed to handle large data sets, convert data, and apply visual techniques to reveal and communicate meaning hidden in data. The available computer tools, however, sometimes impair or limit the ability to display data meaningfully or artistically. Scientific visualization via the computer screen is still in its infancy.

For any given set of data, a variety of correct representations may be able to accomplish the visualization goal. However, there exist even more representations that would not. The correctness of a representation depends on the purpose. A complex,

obscure representation might be quite adequate for personal use, but a general audience will need a more simplified, obvious representation of the data. Again, the goal for any representation is to make information about the data, values, or, the underlying structure clear and immediately obvious to the viewer. It is certainly possible that the best visualization is the one most commonly used with a particular type of data. The best solutions to problems in data visualization result from considering the possibilities and selecting the most appropriate. If the familiar technique best communicates the meaning of the data, then that technique is the most appropriate. Understanding and insight are always the goals.

3.2 Logical Decisions for Windows (LDW) Software

LDW is decision support software created by Logical Decisions, Golden Colorado and is a Microsoft Windows program (Logical Decisions, 1995: 3). LDW helps the analyst rank alternatives that have multiple attributes. The alternatives can be anything that needs to be chosen. Measures describe the alternatives' attributes and are numerical or descriptive variables that capture some quality of the alternatives. The measures are organized under goals. Goals are broad concerns that the choice may affect. The goals and measures are organized into a hierarchy. The broadest goals are at the top, while more specific goals or subgoals are in the middle. Quantitative or descriptive measures are at the bottom. The goals are used to aggregate the scores for the measures and sub-goals beneath them. LDW uses an overall score called utility or value to rank the alternatives. After the analyst defines the alternatives, LDW assists the analyst by

evaluating the decisions quantitatively. LDW provides several different displays to view the results (Logical Decisions, 1995: 175 - 200) and can be found in Appendix A.

3.3 Visualization Shortcomings

The need for creating a new visualization software tool came in direct response to the perceived shortcomings of Logical Decisions for Windows (LDW) . LDW provides decision analysis software support; however, the visualization displays are limited in their ability to provide additional insight into the data set by displaying information in only two dimensions. The main output is the traditional bar chart or histogram or plots in two dimensions. These particular displays limit the ability of the analysts to see underlying patterns or relationships within the data and consequently limit the analysis and interpretation of the results. User interaction with the output is also limited. Providing the user with the capability to dynamically create additional views of the output can greatly increase the understanding and analysis of the data. By not including this ability, the overall analysis could be weakened, or, at best, more time consuming.

3.4 Data Description

This section first provides a general description of the type of multidimensional data being analyzed. Then, a more specific description of the data used in the Paducach WAG 6 value hierarchy model is presented.

3.4.1 General Description

The data used for evaluating the software visualization tool is multidimensional and hierarchical. Measures are located at the bottom of the value hierarchy and describe

the possible alternatives. They are numerical or descriptive variables that capture some quality of the alternatives. Each alternative has a raw score on each measure within the hierarchy. The measures, within the hierarchy, are organized under goals. Goals are broad concerns that hold measures and other goals. Goals do not have a raw score associated with them, but are used to aggregate the values for the measures and sub-goals beneath them. Whereas measures are specific, goals are more general in nature and represent values or objectives of the decision maker. The goals and measures are organized into a hierarchy in a manner where the broadest goals are at the top, specific goals are located in the middle, and the descriptive measures are at the bottom.

3.4.2 Specific Description

The specific data used to validate the software visualization tool was provided by the Paducah Gaseous Diffusion Plant Waste Area Group (WAG) 6 analysis team. A value hierarchy model was developed to determine the appropriate environmental response for a remediation task. At the lowest level of the hierarchy are 28 evaluation measures which are organized under goals. Each evaluation measure has a single dimensional value function. The WAG 6 team decided that single dimensional value functions, across the possible range of values, accurately reflected the evaluation measures. Most of the single dimensional value functions developed were linear with either an increasing or decreasing slope, as appropriate. Also, each alternative will have a raw score for each measure. The raw score is then used as input into the corresponding function. The result is a numerical value of that particular measurement for a given alternative. A weighted-value is computed for each measure by taking the product of the measure's value and its weight.

This product is added to other products associated with the goal above to form an aggregate value of that goal. A requirement for any goal is that the weights of the subgoals or measures that form the linear combination to compute its value must sum to one (see Table 5). For instance, at the highest level of the hierarchy is a single-valued goal representing an alternative's overall score. The value of this goal is computed by taking a linear combination of the values of the five goals.

Table 5. Weight Requirement Example

Alternative (Train) Overall Score - Goal 0		
Goal Number	Goal Name	Goal Weight
1	Long-Term Effectiveness and Performance	1/4
2	Reduction of Toxicity, Mobility, or Volume through Treatment	1/4
3	Short-Term Effectiveness	1/6
4	Implementability	1/6
5	Cost	1/6
	TOTAL WEIGHT	1

The goals consist of either lower level goals and/or evaluation measures. Each goal also has a value and a weight. Again, the weights at any level of the hierarchy must sum to one. There are 23 alternatives to be evaluated and 28 evaluation measures.

3.5 Software Visualization Tool

The software visualization tool was designed to provide an interactive environment for end-user visual data exploration. Visualization techniques are applied to raw data in order to gain insight into the structure of the data itself or the process, procedure, or trend represented by that data. Consequently, any display that yields worthwhile insight into the data set is a valuable, and more importantly, valid representation (Brittain, 1990: 323). Therefore, it becomes quite obvious that a general purpose data visualization system

should support multiple displays with each having its own advantages and being complementary to the others. In any three-dimensional graphics system, interactivity of the movement is crucial to revealing the three-dimensional nature of the visualization. Moreover, when exploring a volume of raw or processed data, interactivity is necessary to reduce the visual analysis time and thereby create a practical tool for the analyst or decision maker to use when exploring and analyzing the data.

3.5.1 General Description

The basic design of this software visualization tool was based on the need to create a tool that could display multidimensional data and provide the user with a certain amount of visual control by interactively manipulating particular aspects of the visual display. By displaying the visual images in three dimensions, additional data can now be represented. The added dimensionality should enable the user to gain greater insight into the data set. User interaction would further extend the ability to see more underlying structure of the data. The user has the ability to rotate and zoom in and out of the visual displays in such a manner that different views can be seen. This added functionality allows the user to explore the data set from a variety of angles and increases the opportunity to see patterns within the data set that may have been previously hidden.

These next sections describe the specific visualization representations that the software visualization tool is capable of displaying. The displays were based upon the various techniques discussed in Chapter 2, such as using glyphs, color, three dimensions, and animation to represent higher dimensional data. The intent is for these displays to

show the data in a format that enables the user to see patterns and relationships that might otherwise be hidden as well as perform sensitivity analysis.

3.5.2 Visualization Displays

The software visualization tool allows the user to select different displays from a selection button panel. Once a display has been selected, the user interface queries the analyst for the required input. After the user has provided the necessary information, the appropriate visualization is displayed. To avoid the problem of having too much data on the screen, an information panel is displayed each time the user selects a display object on the screen. This is accomplished by placing the cursor over a particular object and holding down the left mouse button. The user may turn the information panel off at any time. Specific information about the selected object is provided in the information panel located at the bottom of the display window. The figures showing the three-dimensional scatter plots include labels and a vertical line to identify each of the axes (X, Y, and Z). However, the actual software displays do not currently have this feature implemented.

3.5.2.1 Weight Hierarchy Display

The weight hierarchy display shows the general structure of the data hierarchy. It is a tree-like representation with goals and measures drawn as rectangular, three-dimensional boxes. Each subgoal and measure is placed under its top level goal. When there is more than one measure attached to a goal, the additional measures are displayed as being projected onto the Z axis. This gives the appearance that the measures are placed into the screen. The vertical height (Y axis) and the depth (Z axis) are set at a fixed length. However, the width of each rectangular cube is proportional to its corresponding

absolute weight or contribution to the overall hierarchy value. The display provides the user with a good representation of both the structure of the hierarchy and the relative contribution of each goal and/or measure to the overall value. The display is shown in Figure 5 and is initially displayed when the software visualization tool is started.

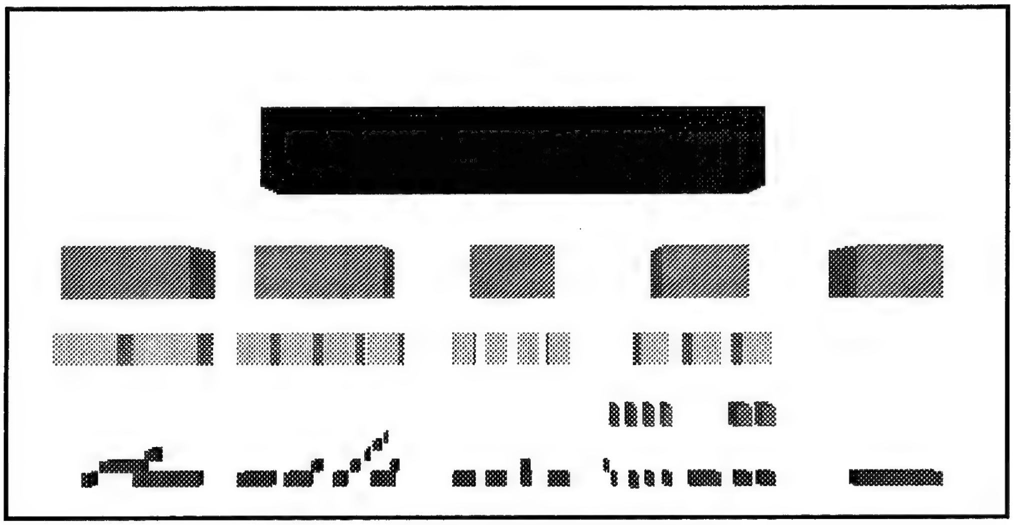


Figure 5. Weight Hierarchy Display

3.5.2.2 Three-Dimensional Scatter Plot

A scatter plot is a basic graphing technique, normally with two axes (X and Y), that shows the relationship between two variables, with the pairs of values represented by points or other symbols. However, using three dimensions enables additional information to be plotted. As previously mentioned, Logical Decisions for Windows (LDW) only provides the user the capability to plot two variables using two dimensions.

Scatter plots are still an effective means of displaying data. With the added dimensionality of the three-dimensional scatter plot and color coding, the user can dynamically plot any three of the five criteria goals (Long-Term Effectiveness and Permanence, Short-Term Effectiveness, Implementability, Reduction of Toxicity, Mobility

or Volume Through Treatment, and Cost) or the CERCLA value. The color coding is assigned as follows from highest value to lowest value:

- Blue: The highest CERCLA value of all 23 alternatives
- Cyan: 2 - 6
- Green: 7 - 12
- Yellow: 13 - 17
- Orange: 18 - 22
- Red: The lowest CERCLA value of all 23 alternatives

As stated in Section 2.7.3, there is no immediate ordering scheme. The color blue was chosen to represent the highest valued alternative as it is generally regarded as reliable and is very familiar to the eye. Red was chosen to represent the lowest valued alternative because this color is generally accepted to represent caution or danger. This makes the color red a good choice to alert the user of the lowest valued alternative. To represent the entire range of values, the natural order found within the light spectrum was used to assign a color to a specific interval within the range. The entire range, from the lowest value to the highest value, is represented by red, orange, yellow, green, cyan, and blue.

The three-dimensional scatter plots provide more information on a particular display compared to the two-dimensional plots and enable the user to see more patterns and structure within the data set. The software visualization tool currently displays a cube for each of the 23 alternatives. The values of the five criteria goals and the CERCLA value can then be arbitrarily assigned to any of the three axis (X, Y, Z). Figure 6 shows an example of the three-dimensional scatter plot.

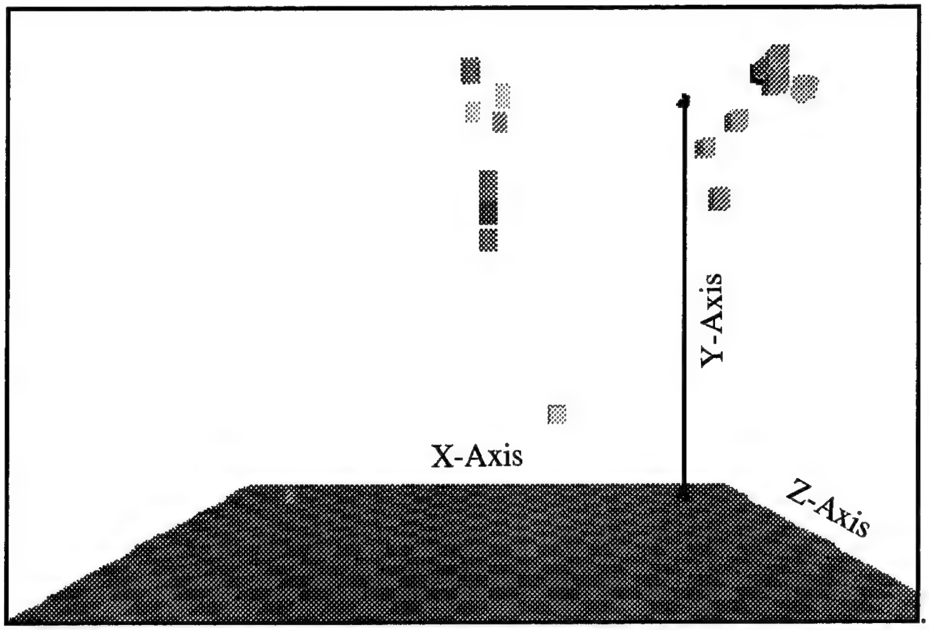


Figure 6. Three-dimensional Scatter Plot

3.5.2.3 Advanced Three-Dimensional Scatter Plot

In addition to utilizing the additional axis for increasing the dimensionality of the display, glyphs can be combined for an even higher dimensional representation. Glyphs, graphical objects whose attributes such as position, size, shape, color, orientation, etc., can be incorporated into the visual representation in order to display higher dimensional data (Ribarsky, 1994: 57). For instance, plotting a three-dimensional cube as a glyph enables the object to represent multiple attributes of the data. This increases the display's dimensionality to six. The values of the five criteria goals and the CERCLA value are represented visually using the X, Y, and Z axis, along with the height, width, and depth of the cube. The user interactively assigns a particular criteria or overall value to any of the six dimensions. The color of the cube represents its overall value.

The glyphs or displayed rectangular cubes for all 23 alternatives enable the user to quickly identify patterns and clusters within the data with respect to the five criteria goals

and the CERCLA value. The 23 alternatives are neatly displayed, capable of showing six different attributes of the data. The amount of clutter is reduced, allowing the user to focus on the actual visualization representing the data set. An example of the advanced scatter plot can be seen in Figure 7.

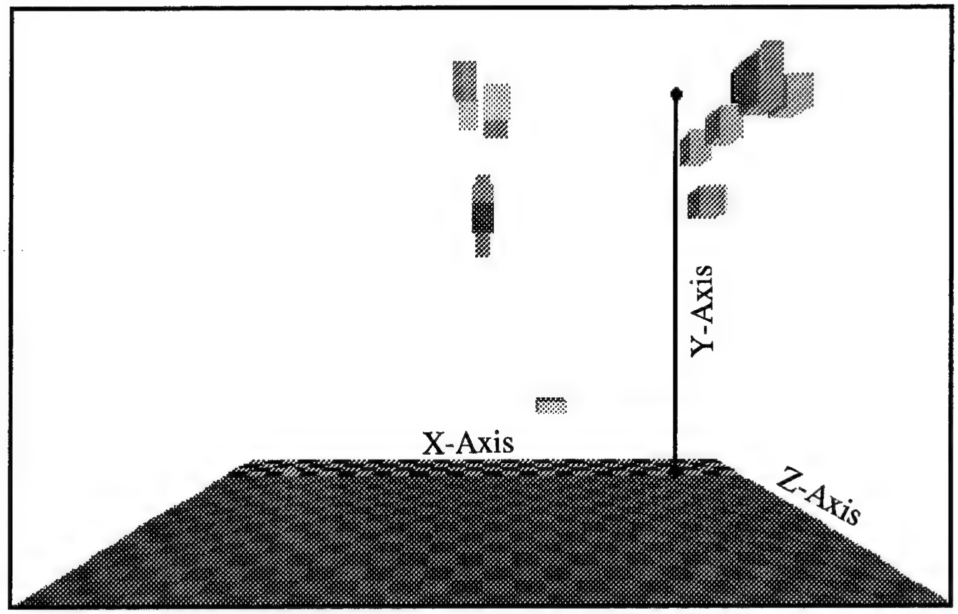


Figure 7. Advanced Three-dimensional Scatter Plot

3.5.2.4 Goal Display

The goals are each displayed as a three-dimensional rectangular cube. The rectangular cube is displayed associating its vertical height to represent the goal's value. The user can select any combination of goals against any combination of alternatives. The representation is also color coded, by associating the color of the cube to its corresponding weight value.

Having the capability to view the entire data hierarchy enables the user to search for patterns or hidden relationships within the data set. This particular display visually

enforces comparisons of the data. In addition, the display shows differences and variation among the goals with respect to the selected alternatives. By using the zoom and rotation display options, the user can see the entire data set and more importantly how the data varies. Although the scatter plots just display information about the top criteria goals, this display allows the user to search further into the goal hierarchy as shown in Figure 8.

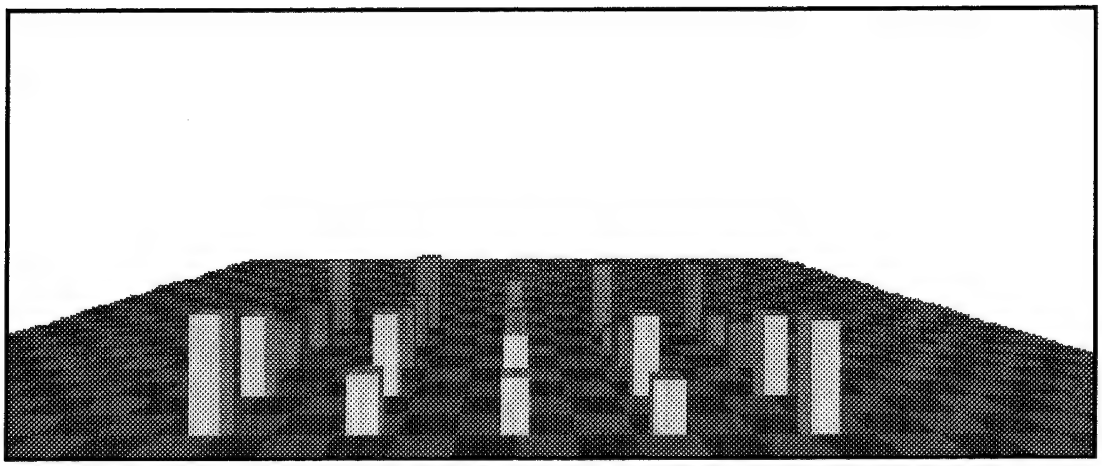


Figure 8. Goal Display

3.5.2.5 Sensitivity Analysis Goal Display

This display provides the user with the opportunity to consider many different scenarios. The user may select from one to all 23 alternatives to display. For any one of the five criteria, the weight may be adjusted to view what the new results would be. Each alternative is represented by a rectangular cube. The height of the rectangular cube is determined by the overall CERCLA value computed by using the new weights provided by the user. The user also has the option to display up to five different rows at one time. The user can then determine how sensitive the overall CERCLA value is to the weight. For each row, the alternative with the highest overall CERCLA value is color coded blue.

The alternative with the lowest value is color coded red. For the first row, the remaining alternatives are color coded green. For each subsequent row, if the overall value increases or stays the same, the rectangular cube remains green. If the value decreases, the color becomes yellow.

The user can easily and very quickly identify the best and worst selections and more importantly how the change impacts the entire data set. The user can immediately know what alternative is the best choice based on changes to the criteria goals' weights and whether or not certain alternatives would ever produce the best value. An example is shown in Figure 9.

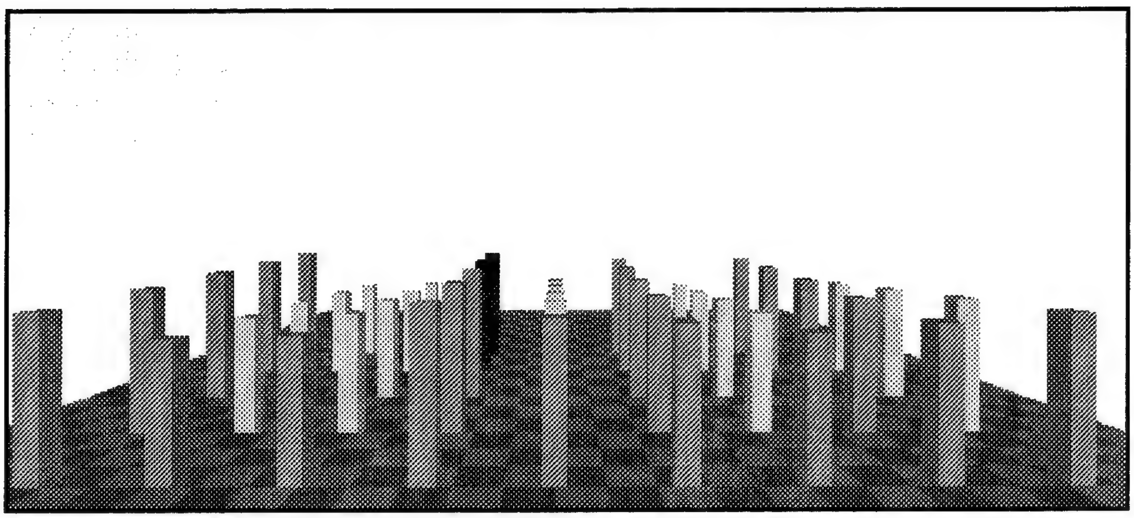


Figure 9. Sensitivity Analysis Goal Display

3.5.2.6 Measure Display

The measures are each displayed as a three-dimensional rectangular cube, associating its vertical height to represent the measure's value. The user can select any combination of measures against any combination of alternatives. The representation is

also color coded, by associating the color of the rectangular cube to its corresponding weight value.

At the heart of the data set are the measures, which along with the weights, drive the overall goal value. It is extremely important to be able to view the entire lowest level tier of the data hierarchy, searching for patterns or hidden relationships within the data set. This particular display visually enforces comparisons of the data. In addition, the display shows differences and variation among the measures with respect to the selected alternatives. Small multiples reveal, all at once, a scope of alternatives and a range of options (Tufte, 1991: 56). By using the zoom and rotation display options, the user can see the entire data set, and more importantly how the data varies. Measures that are color coded with a high weight, but whose value are low, have the greatest potential for increased contribution to the CERCLA value. The user can easily search the display, identifying those measures. Figure 10 provides an example of the measure display.

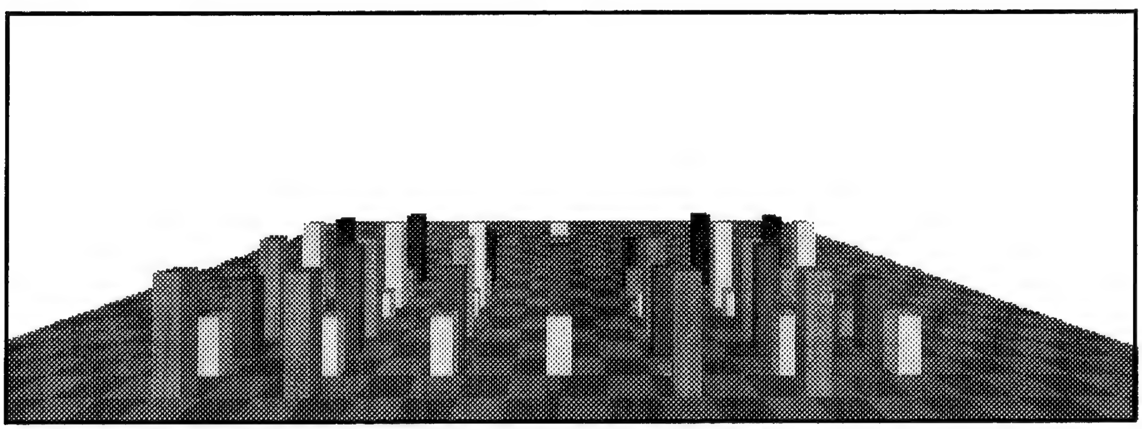


Figure 10. Measure Display

3.5.2.7 Animated Alternative Display

An animated display is created by rapidly changing still images to create the illusion of movement. The user can select one or two alternatives to be displayed. The first alternative is represented by a three-dimensional cube and the second by a three-dimensional sphere. The user then assigns a color to each object. The objects appear to "travel" across the screen. The actual path of the cube and or sphere is determined by associating the value for all 28 measures of a specific alternative to virtual points on the screen. As the objects move across the screen, additional cubes and or spheres remain on the screen at the 28 discrete points representing the value of a measure at that particular point. The user is left with a plot of the two alternatives and can observe the display for any erratic behavior. This provides an easy way to compare two alternatives against all 28 measures as seen in Figure 11.

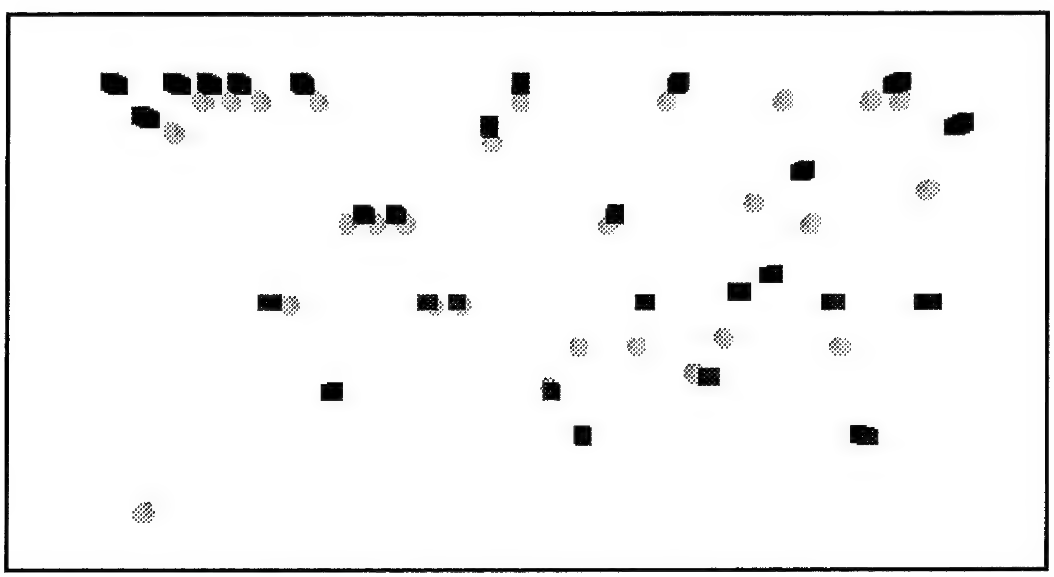


Figure 11. Animated Alternative Display

3.5.2.8 Animated Measure Display

Although similar to the animated alternative display, the actual path of this display is determined by associating each of the 23 alternative's values of a specific measure to virtual points on the screen. The user can easily observe all values for a specific measure and notice any variability within that particular data set. The display is capable of displaying one or two measures by using different shapes to represent the different measures. The user can then compare and contrast any differences in the respective measures. The first measure is represented by a three-dimensional cube and the second by a three-dimensional sphere. Similar to the animated alternative display, the user assigns a color to each object and as the objects move across the screen, additional cubes and/or spheres remain on the screen at the 23 discrete points representing a particular alternative. The user is left with a plot of the two measures and can observe the plots for any unusual behavior. This provides an easy way to compare two measures against each other for all 23 alternatives. When a display object is selected, the corresponding measure is also highlighted and the information panel is shown. Figure 12 provides an example of the animated measure display.

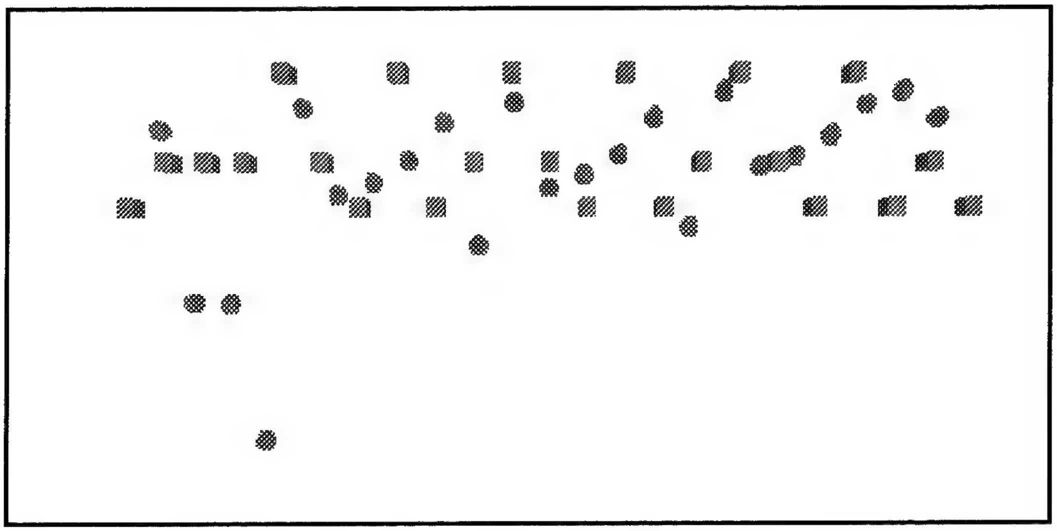


Figure 12. Animated Measure Display

Animation can be an extremely effective technique for understanding complex three-dimensional scenes. Animation greatly reduces the cognitive effort of the user by providing object constancy (Robertson, 1993: 61). The user no longer has to focus on 28 individual static objects, but just has to observe the motion. Animation can help to sort out the relevant data from a large amount of raw numbers and to show variations within the data set. The greatest advantage of this technique is that it provides a quick, almost simultaneous look of the entire data set with respect to a specific measure or alternative.

3.6 Promising Approaches

Although this field of study is still an emerging technology, the future offers great potential. There are several additional visualization techniques or displays that might greatly enhance an analysis. However, these particular techniques or displays were unable to be implemented into this particular software visualization tool. The main reasons for them not being included at this time are the associated developmental costs, the amount of time needed for development, and the required knowledge to incorporate them.

3.6.1 Virtual Reality

This is a fast growing field of study with respect to information visualization. Virtual reality has been associated with almost everything to do with computers and images. At the forefront of any virtual reality system is a three-dimensional virtual environment. This can be based upon some concept design, can represent a physical object or environment, or can even incorporate using some abstract data set. The actual nature of the data is not important; what is important is the way the data is viewed and ultimately manipulated.

Conventional computer visualization techniques separate the viewer from the image by creating it upon a flat display screen. However, the idea of virtual reality is to place the user "inside" the data by using a head-mounted display or other appropriate technique. By tracking the position and orientation of the user's head, a head-mounted virtual reality system computes and displays corresponding views of the virtual environment in real time. This provides the user with a sensation of being immersed within the environment and the data set itself.

Although a three-dimensional scene can be represented as an image being displayed upon a two-dimensional screen, the scene remains flat regardless of the how the user's head is moved. Immersive virtual reality eliminates this constraint and allows the user to explore any view of a virtual environment according to the user's position and orientation. This is the way people explore the real world, so it becomes a natural way of exploring virtual worlds. The visual representations no longer have to be rotated or moved; the user "moves" about the data. This enables the user to more easily explore the data set and find more meaningful information.

In addition to obtaining better views of the data using virtual reality techniques, the interaction between the user and the display is greatly improved as well. With the use of interactive gloves, a user can "touch and manipulate" virtual objects. If the right level of system support is available, the user can interact with the actual software that is creating the image. This interaction allows the user to then influence any dynamic parameter created. These parameters could include color, texture, lighting, and geometry.

Virtual reality systems are being developed for visualizing and exploring data sets, virtual objects, and environments. Because the user is able to interact with the images, a virtual reality system has particular advantages, especially when interfaced to a simulation being controlled by the user. Virtual reality is still a very new technology; consequently, there are limited off-the-shelf software products that can be applied to specific visualization applications.

3.6.2 Sound

There is a great deal of interest in incorporating sound more fully into computer displays and animation as a way of providing additional information to the viewer. Typically, sound has been used in computer applications to provide feedback about an error condition or when the computer has completed a task. Sound has also been applied to signal an alert about a change in the current situation, such as arrival of electronic mail or a potentially dangerous situation being monitored by an application.

With respect to data analysis without sound, the addition of sound enables the user to take advantage of two of the human senses and, if used properly, increases understanding (Brown, 1995: 140). One possible application of incorporating sound into

the visual display is for the data to control both the image displayed and the sound heard. A second technique is for the sound to provide additional information, increasing the dimensionality that can be represented. Sound in virtual environments help users get oriented with their surroundings and improve upon the virtual world's perception of reality. The virtual environments are more comfortable and feel more complete.

3.6.3 Visualization Tools

One of the largest problems facing anyone selecting visualization software and an appropriate computer is the wide choice available. On-site demonstrations, talking with people who share the same problems, talking to user groups, and visiting academic institutions that specialize in visualization can assist in gaining the necessary information to help make the correct choice. Visualization products for personal computers and workstations are available from a number of companies. Table 6 lists available software to perform scientific, engineering, and business applications (Gershon, 1993: 148):

Table 6. Visualization Software

Title	Vendor
AVS, AVS Animator, & Developers AVS	Advanced Visual Systems, Inc.
COVIS	CoHort Software
DADISP	DSP Development Corp.
Data Desk	Data Description, Inc.
Data Visualizer	Wavefront Technologies, Inc.
DICER & Transform	Spyglass, Inc.
EASY5X	The Boeing Co.
IDL	Research System, Inc.
IPLAB Spectrum	Signal Analytics Corp.
Maps&Data	MapInfo Corp.
PVWAVE Command Language	Visual Numerics
Sliders and Dials	Golden Technologies
Synchroworks	Oberon Software, Inc.
Tecplot	Amtec Engineering, Inc.
Visualization Data Explorer	IBM Research
Voxelbox	Jaguar Software

3.7 Experimental Procedure

As previously mentioned, there are no established techniques for evaluating the effectiveness of a software visualization display. However, in order to validate the software visualization tool created, a validation process is required.

3.7.1 Validation Process

The chosen validation group, as discussed in Section 1.5.2, compared the visual outputs of a currently used software application, Logical Decision for Windows (LDW), with those of the newly created software visualization tool (SVT). One way to find out if the software visualization tool increased the effectiveness of the analyses and provided the users with additional insight into the data was to ask them. A survey, shown in Appendix C, was constructed and provided to each member of the validation group to solicit his

responses. Once the surveys were completed, the data was collected and recorded. Then an analysis of the data was performed, comparing the responses for each visual display.

3.7.2 Data Collection

The survey was given to each member in the validation group in order to obtain his responses with respect to questions concerning the visual displays. The survey questions were developed such that the responses would provide information into which displays were perceived to be more effective in displaying the data, showing the relationships and patterns within the data, and thus providing greater insight into the analysis. The responses to the questions have a numerical range (from 1 to 7) that can be evaluated quantitatively. Also, the users were asked qualitative questions with respect to each display concerning what other uses the display might serve and any suggested improvements. The responses were recorded and maintained in a spreadsheet format for evaluation.

3.7.3 Data Analysis

The surveyed information was analyzed, comparing the responses given for each set of visual displays. Statistical analysis techniques were performed on the responses to the survey to determine if one method of displaying the data was found to be preferred over the other. More importantly, the responses were analyzed to determine if the officers had a better understanding of the data and hence would be able to provide a more accurate analysis of the data using the new displays of the data set. The results of the data analysis are presented in Chapter 5, Findings and Analysis.

3.8 Summary

As scientific data visualization emerges from its infancy, pretty pictures are not enough to accurately portray the data. If the ultimate goal of data visualization is to aid scientists with the exploration of their data and the discovery of new facts, then certain types of visually dazzling presentations may actually be counterproductive (Reuter, 1990: 401). Even less complex visualizations sometimes use colors or symbols that make it difficult to accurately interpret the data. Increasingly, creators of data visualizations need to be sensitive to what types of presentations will enhance as well as complement the perceptual skills of the scientists or analysts who are the intended users of the data visualization system.

The power and usefulness of data visualization is due largely to the strength of human perception. The human visual system is the major player in visualization, but the other senses are starting to be applied to visualization tasks as well. When statistical or scientific data is presented in a plot or other graphical display, the viewer is asked to use his or her visual perception to make quantitative judgments about the display, and to compare relative sizes, locations, orientations, colors, densities, and textures of the elements of the display. Rarely does the designer of the display take into account the fact that human visual perception is a very complicated and subjective process, and that the effectiveness of the display for conveying objective understanding hinges crucially on a wide range of subtle factors, only some of which are under the control of the person making the display (Tukey, 1990: 401 - 402). As data visualization becomes more and more widespread in science, because both today's computer hardware and software make

it easier to produce pictures, it becomes more and more urgent to study these issues of visual perception in order to choose types of displays that avoid the worst pitfalls and convey the relevant information as objectively and effectively as possible.

To complicate matters further, different people may have different perceptions of the same picture, depending on physiology but also depending on what a particular user is looking for and prior experience. It is unlikely that psychologists and graphic designers can provide a complete inventory of all the factors and interactions that can affect visual perception. Even if they could predict an entire inventory, it would still be a difficult task to determine the extent to which those factors might come into play in a particular display of a given set of data. Fortunately, many important principles of good data display don't depend on a deep understanding of psychology, but on just plain common sense and a little thinking about the problem (Reuter, 1990: 402).

4. Software Methodology

This chapter first presents the software challenge, including guidelines for well-engineered software. Next, some general concepts of software engineering to include software requirements, design, and testing are discussed. Then the actual software engineering methodology, with respect to analysis, design, implementation, testing, and validation, applied in creating the software visualization tool is presented. A model of the entire system can be seen in Appendix F.

4.1 Introduction

Like all engineering disciplines, software engineering is not just about producing products but involves producing products in a cost-effective way. One definition of software engineering is "the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines" (Pressman, 1992: 23). Software engineering methods provide the technical approach for actually building the software. These methods include a wide variety of tasks to include project planning and estimation, system and software requirements analysis, design of data structures, overall program architecture and algorithm procedure, coding, testing, and eventually maintenance. Software engineering tools provide automated support for the methods. It is best when the tools are integrated so that information created by one tool can be used by another. Finally, software engineering procedures provide the links that bind the methods and tools together, enabling rational and timely development of computer software. The overall software engineering methodology or approach is chosen

based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. Given unlimited resources, the majority of software problems can probably be solved.

4.2 Software Objective

The software engineering objective was to determine a methodology for building the software visualization tool. The approach developed was based primarily on an object-oriented methodology using Rumbaugh's Object Modeling Techniques (Rumbaugh, 1991: 16 - 18) and traditional software engineering goals and principles.

4.3 The Software Challenge

The challenge for most software engineers is to produce high quality software with a limited amount of resources and to deliver a product constrained to a predetermined time schedule. In order to meet the challenge, the software should be constructed in an appropriate manner.

4.3.1 Well-Engineered Software

Providing the software meets all of the functional requirements, there are four main points which a well-engineered software system should possess (Sommerville, 1992: 3 - 4):

- **The software should be maintainable.** As software is subject to change, it should be written and documented so that changes can be made without unnecessary costs. The component should be designed and implemented so that it can be modified or even reused in many different programs.

- **The software should be reliable.** This implies that the software should perform as expected by users and should not fail more often than is allowed for in its software specification.

- **The software should be efficient.** This particular characteristic does not mean to maximize efficiency at all costs which could actually make the software more difficult to change. Efficiency should be achieved such that a system should not make wasteful use of system resources such as memory and processor cycles.

- **The software should include an appropriate user interface.** The user interface should be designed with the user's capabilities and knowledge. In order to provide its full potential, the user interface must be easily understood.

Attempting to optimize all of these attributes is extremely difficult as some are mutually exclusive and all are subject to the law of diminishing returns. Frequently, there are tradeoffs between attributes based on the system and the needs of the user. Maintainability is usually a key attribute because the majority of software costs are incurred after the software has been put into use.

4.3.2 Software Model

In order to "view" the system being developed, a model can be created using techniques such as those developed by James A. Rumbaugh (Rumbaugh, 1991: 15 - 48). A model is an abstraction of something for the purpose of understanding. A model omits nonessential detail and is easier to manipulate than the original entity. To help in the development of software systems, a model can be constructed to capture different views of the system and more importantly, to verify that the model satisfies the requirements of the system. Gradually, more detail is added to transform the model into an implementation.

4.4 Software Requirements

Software requirements should identify what the system should do without specifying how it should be done. The requirements should be stated so that the design may be validated. A software requirements definition is an abstract description of the services which the system is expected to provide and the constraints under which the system must operate. The requirements definition should only specify external behavior of the system and should not be concerned with design characteristics. Developing software requirements focuses attention on software capabilities. As the requirements definition is further developed, a better understanding of the user's needs is achieved. However, this can cause the perceived requirements to change. The time required to analyze requirements and to develop a system can then take much longer than originally expected. The possibility of change should be anticipated early on so that major changes required to the system can be minimized.

4.4.1 Requirement Categories

Requirements typically fall into two categories: functional system requirements and non-functional system requirements. Functional system requirements are system services which are expected by the user of the system. The user is usually uninterested in how these services are implemented, just that the required functionality is met. The non-functional requirements set out the constraints under which the system must operate and the standards which must be met by the delivered system. Although both functional and non-functional requirements are subject to change, non-functional requirements are particularly affected by changes in hardware.

4.4.2 Requirements Analysis

Requirements analysis is concerned with devising a precise, concise, understandable, and correct model of the real-world system. The purpose of an object-oriented analysis is to model the real-world system in such a manner that it can be understood. The end result of this analysis should be a thorough understanding of the problem in order to prepare for a complete design of the system. According to Rumbaugh, the analysis model is a precise, concise representation of the problem that permits answering questions and building a solution (Rumbaugh, 1991: 149).

4.4.3 Object Model

The object model identifies relevant classes to be included as part of the overall system (see Appendix D). Objects include physical entities as well as concepts. An object class describes a group of objects with similar properties or attributes, common behavior or operations, common relationships to other objects, and common semantics. Objects in a class have the same attributes and behavior patterns. An attribute is a data value associated to an object belonging to a particular class. An operation is a function or transformation that may be applied to or by objects in a class. All objects in a class share the same operations. Most objects are unique with respect to the differences in their attributes values and their relationships to other objects (Rumbaugh, 1991: 152 - 169).

4.5 Software Design

As soon as the analysis of the system and software is completed, the design phase of the software life cycle will usually begin. Design is a creative process which requires experience and some creativity on the part of the designer. Design is an art that is

developed from practice, experience, and the study of existing systems. It is a process that cannot just be learned from a book. However, good design is one of the keys to successful and effective engineering.

4.6 Object Design

The analysis phase determines what the implementation must do and the system design phase determines the overall plan to accomplish that. The object design phase determines the specific plan of attack. This phase determines the full definitions of the classes and associations to actually implement the software. Also, the interfaces and algorithms of the methods used to implement operations are constructed. The strategy selected during system design is carried out; however, much more detail is added. Most noticeable during this phase is a shift from domain specific concepts toward computer concepts. The objects and classes discovered during analysis serve as the building blocks of the actual design to be implemented. The goal is to choose among different strategies to implement them while always considering execution time, memory, and other measure costs. The classes, attributes, methods, and associations determined during analysis must now be implemented as specific data structures. Although optimization of the design should be sought, it should not be carried to excess. Throughout the design phase, ease of implementation, maintainability, and extensibility are always important concerns.

4.7 Software Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. It is not considered unusual for a software development organization to expend 40 percent of the total project effort on

testing. In extreme situations, such as software associated with human life, the testing effort itself can cost three to five times as much as all other software engineering steps combined (Pressman, 1992: 595).

During earlier developmental phases, the software engineer attempts to build software from an abstract concept into a full, working implementation. However, during testing the engineer creates a series of test cases that are intended to "crash" the software that has been built. Oddly, testing is the one phase of the software development life cycle that could be viewed as destructive rather than constructive. The objectives of testing may appear somewhat different than what might be expected and are provided by Pressman (Pressman, 1992: 596):

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding a currently undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

The objective is to design tests that systematically uncover a variety of errors, but do so with a minimum amount of time and effort. If testing is conducted successfully, it should uncover errors in the software. In addition, testing helps demonstrate that the software functions appear to be working according to specification and that performance requirements appear to have been met. Testing can also provide possible insight into the reliability and quality of the software as a whole. Unfortunately, there is one thing that testing cannot do: "Testing cannot show the absence of defects, it can only show that software defects are present" (Pressman, 1992: 597).

4.8 Analysis Process

A software requirements analysis was conducted to determine the actual requirements of the software visualization tool. Potential users of the tool and thesis committee members determined the functional and non-functional requirements. The requirements were based primarily on the shortcomings of currently used analysis software as discussed in Section 3.3. Once the requirements were established, a software class listing was constructed, identifying the relevant classes (Appendix D).

4.8.1 Functional Requirements

With respect to this software visualization tool, the functional requirements are:

- Read in "raw" data from some external file source.
- Read in "computed" data from some external file source.
- Compute the lowest-level measurement scores from "raw" data.
- Compute the values at all levels within the hierarchy.
- Allow the user to change the weight of a goal or measurement.
- Ensure the linear combination of weights used to compute a goal's value sum to one.
- Perform two-way sensitivity analysis.
- Display the goal or measurement scores at all levels within the hierarchy.
- Display the sensitivity analysis results.

4.8.2 Non-functional Requirements

The specific non-functional requirements of the software visualization tool are:

- The software application will run on a personal computer (PC).

- The operating system will be Windows 95.

4.8.3 Relevant Classes

The software visualization tool uses several class structures. A complete listing of all classes used in the implementation, including language and vendor supplied, are found in Appendix D. The specific classes built for this software visualization tool consisted of the goal and measure classes. The measure class was implemented as a subclass of the goal class and therefore inherits all the attributes and methods of the goal class.

4.9 Design Process

The problem to be solved was to create a software visualization tool to visually display the data set discussed in Section 3.4.. The analysis phase determined the system and software requirements. Based on those requirements and the relevant classes identified, an overall system architecture and object model was constructed (Appendix F).

4.9.1 System Architecture

The overall system was divided into subsystems to form the overall architecture of the system. The software visualization tool was to accept input from an external data file (Appendix G), which was created by Logical Decisions Decision Support Software. The first subsystem of the software application was concerned with data input and was designed to read the data from that exported file. The next subsystem consisted of the methods to transform the raw data into utility values. The final subsystem consisted of actually displaying the data.

4.9.2 Object Design

Using the classes identified during analysis, an object model was constructed to represent the overall system architecture and is shown in Appendix F. This model shows the classes used and the associations between those classes within the overall structure. This model was used to determine the required data structures and served as the foundation for the actual implementation.

4.10 Implementation

Once the overall framework was established, the implementation phase began. The object classes and relationships developed during the design phase were translated into a particular programming language. During the design phase, a programming language was selected to implement the design. After considering several possibilities, the language chosen was Java.

4.10.1 Java Programming Language

Java was chosen because of its portability, its user interface, and its three-dimensional programming capabilities. Java is an object-oriented programming language developed by Sun Microsystems. The Java language was designed to be small, simple, and portable across platforms and operating systems, both at the source and at the binary level. This makes Java a good choice because Java programs can run on any machine that has the Java virtual machine installed or a "Java enabled" browser, such as Netscape Navigator and Microsoft's Internet Explorer. Java was written as a full-fledged, general-purpose programming language capable of accomplishing the same sort of tasks and

solving the same sorts of problems that other programming languages can, such as C or C++.

4.10.1.1 Java Applet

Java applets run and are displayed inside a Web page. This special feature makes them even more portable than a regular application. The software visualization tool was designed and implemented as a Java applet. Some basic activities that are happening during execution are actually being monitored by the browser so when an event occurs the corresponding method is invoked. Each of these methods can then be overwritten with an applet's specific instructions. There are five major applet activities (Lemay, 1997:162-163):

- **Initialization.** Initialization occurs when the applet is first loaded. The initialization sets up an initial state. The user interface buttons of the software visualization tool are created in this method.

- **Starting.** After an applet is initialized, it is started. Starting is different because it can happen many different times during an applet's lifetime; however, initialization happens only once.

- **Stopping.** Stopping and starting are associated with each other. Stopping occurs when the page that contains the running applet is either left or by calling the method directly.

- **Running.** This is where Java actually runs the applet.

- **Painting.** This is the way an applet actually draws something on the screen. Painting can occur many thousands of times during an applet's life cycle.

4.10.1.2 Java 3-D API

Sun's Java 3-D Application Programming Interface (API) is not currently available. The scheduled release is March 1998. In order to create the three-dimensional

views in the software application, a Java 3-D API from InWorld VR, Inc was used. A free of charge, "limited evaluation API" was used in this development. However, several classes were not included with this version. Most noteworthy was the absence of the line class, which diminished the overall quality of the displays created.

4.10.2 Programming Environment

Sun Microsystem's Java Development Kit for Windows 95, version 1.1 was used to build the software visualization tool. All programming was done on a personal computer with a 486 (DX2/66) microprocessor.

4.11 Testing Process

The testing process consisted of verifying that each method and procedure built performed its required task. The methods were isolated and tested independently of the entire system. Once each component successfully passed its test, the components were integrated into their respective subsystem. The subsystems were tested to ensure they performed their functionality. Finally, all of the subsystems were integrated into the complete system and system testing was performed. The system testing was performed by actually using the software visualization tool and verifying that the correct information was being stored in the objects and that the displays were generated.

4.12 Validation Process

The software validation process is discussed in Section 3.7.1. The actual findings of the validation are presented in Chapter 5, Results and Analysis.

4.13 Summary

The underlying methodology used in creating the software visualization tool was incorporating modular programming with an object-oriented approach. The overall developmental approach was decomposed into different phases. The phases were software analysis, design, coding and implementation, testing, and validation. The goal was to create a reliable software system that would be flexible, maintainable, and capable of evolving to meet changing needs.

5. Results and Analysis

This chapter presents the results of the software validation. First, the experiment and all assumptions made are discussed. This is followed by a description of each visualization comparison made and the statistical findings. Then an overall assessment is presented. In addition to the analysis results, experimental problems encountered during the validation process are addressed.

5.1 Introduction

The validation group reviewed visualization displays generated by Logical Decisions for Windows (LDW) and the software visualization tool (SVT). Both software applications used the data discussed in Section 3.4. The validation group consisted of six members who had taken Decision Analysis (DA) courses and six members with no experience in this field of study. Of the six who had DA experience, five had used or were familiar with the LDW software package. The actual displays generated by LDW and the software visualization tool during the validation process can be seen in Appendix A.

5.2 Validation Problems

There were several problems associated with the validation process.

- There were not enough "experts" to evaluate the software. Only twelve people were able to evaluate the software and of them only six had DA experience. For a statistically valid study at least 30 subjects should be used.

- A computer room was reserved to conduct the validation; however, students not participating in the validation process interrupted the process several times. Better control of the test area is needed.
- More time was needed to enable the individuals to familiarize themselves with both software packages and to become more aware of the tools' uses.
- The software visualization tool being validated was not completely ready for validation. Proper labeling of an object selected and providing the object's corresponding information to the computer screen had not been fully implemented.

5.3 Statistical Assumptions

Although the number of observations was well below the minimum desired amount of 30, the data will be assumed to be normally distributed. All statistical evaluations are performed using an alpha value of 0.05. For the 21 comparisons made, an F-test was performed on each group of data to verify that the variances for the responses within each comparison were statistically equal. The data set was evaluated to determine if there was any significance between the individuals who had DA experience and those who did not. Statistically, there was no difference in the results. However, the members with DA experience provided more insightful and detailed written comments concerning the displays which will be included in the overall assessment section for each comparison.

5.4 LDW Displays

The following LDW displays were used in the validation comparisons (Logical Decisions, 1995: 175 - 200):

- **Bubble Diagram:** Shows a network of lines and circles that show which measures were selected to be compared and the relative weights of each.

- **Goals Hierarchy:** Shows the goals and measures in the analysis in a structure like an organization chart.

- **Rank Alternatives:** Shows a ranking of the alternatives based on any of the goals or measures.

- **Stacked Bar Ranking:** Similar to the rank alternatives, but provides more detail on how the alternatives' values on the lower level goals and measures affect the value on the selected goal.

- **Dynamic Sensitivity:** Displays the effects of changes in the weights for the goals and measures.

- **Sensitivity Graph:** Displays the effect of varying a measure or goal's weight from 0 to 100 percent.

- **Sensitivity Table:** Displays an overall ranking based on any percentage of weight on a goal or measure.

- **Scatter Diagrams:** Compares the performance of the alternatives on any two measures or goals. One measure or goal may be represented on either the X or Y axis.

- **Graph an Alternative:** Displays a bar graph showing the performance of a single alternative on the measures or goals.

- **Compare Alternatives Graph:** Displays a comparison of the differences between two alternatives as a graph.

5.5 Results

A paired t-Test was conducted on each comparison group to determine if there was a statistical difference in the mean responses between the LDW and SVT software

displays. The test for each comparison was based on the null hypothesis that the two means were equal. Therefore, whenever the absolute value of the t statistic is greater than the t critical value, the null hypothesis is rejected and it is concluded that the means are statistically different. Whenever the t statistic is less than or equal to the t critical value, the null hypothesis is not rejected and the means are concluded to be statistically equal. For each software visualization tool display, there is a description of each comparison made and then an overall assessment. Appendix B contains the following supplemental information: tables listing the means and variances for each software display, the t statistic value, the p-value, and the t critical value for each comparison, and two histograms for each comparison, plotting the frequency of the actual responses.

5.5.1 Weight Hierarchy Display

The SVT weight hierarchy display was compared to the LDW bubble diagram and the LDW goals hierarchy diagram. The question dealt with how effective the displays were in representing the overall structure of the data set and the relative contribution of each goal, subgoal, and evaluation measure to the overall CERCLA value.

5.5.1.1 Comparison 1-1

Comparison 1-1 was between the LDW bubble chart and the SVT weight hierarchy display. The paired t-Test identified the two means to be statistically different. The mean response for the bubble chart was 4.67 and the mean response for the weight hierarchy was just over 5.42 with an associated p-value of 0.043.

5.5.1.2 Comparison 1-2

Comparison 1-2 was between the LDW goals hierarchy diagram and the SVT weight hierarchy display. The paired t-Test showed the mean responses to be statistically different. With respect to this comparison, the SVT weight hierarchy mean response was significantly higher. The mean for the LDW goal hierarchy was 4.58 and the mean response to the weight hierarchy was 6.17 with an associated p-value of 0.00037.

5.5.1.3 Weight Hierarchy Assessment

In both comparisons, the SVT weight hierarchy was shown to statistically have a mean greater than either of the responses to the LDW displays. Based on the higher mean, it appears as though the SVT weight hierarchy was perceived to be more effective than either of the LDW displays in showing the overall structure of the data set and the relative contribution of each goal and measure to the overall CERCLA value. The weight hierarchy mean increased with the second comparison and its variance reduced dramatically. The mean for both LDW displays stayed approximately the same between both comparisons; however the variances changed significantly as well.

Specific comments mentioned that the SVT display showed mutual exclusivity as well as relative value better. One member thought it could be used for instructing students in how DA/value hierarchies work and giving a visual intuitive feel for value hierarchies. However, it was noted from one validation member that the resolution of the picture made it difficult to tell the actual size of the three-dimensional cube and that possibly using just two-dimensions would be better for this display.

5.5.2 Three-Dimensional Scatter Plot

The SVT three-dimensional scatter plot was compared to the LDW rank alternatives display and the LDW scatter diagram. The question dealt with how effective the displays were in representing the influence of each of the five criteria goals with respect to the overall CERCLA value.

5.5.2.1 Comparison 2-1

Comparison 2-1 was between the LDW rank alternatives and the SVT three-dimensional scatter plot display. The paired t-Test identified the two means to be statistically equal. However, the three-dimensional scatter plot mean was 5.42 as compared to the rank alternative mean of 4.42. The associated p-value was 0.16.

5.5.2.2 Comparison 2-2

Comparison 2-2 was between the LDW scatter diagram and the SVT three-dimensional scatter plot display. The paired t-Test identified the two means to be statistically different. The three-dimensional scatter plot mean was 5.33 as compared to the scatter diagram mean of 4.25. The associated p-value was extremely low at 0.01.

5.5.2.3 Three-Dimensional Scatter Plot Assessment

The SVT three-dimensional scatter plot display was shown to statistically have a mean greater than that of the scatter diagram. The comparison between the three-dimensional scatter plot and the scatter diagram had a more direct mapping, whereas the comparison between the rank alternatives display and the three-dimensional scatter plot did not. Oddly, though, the mean for the three-dimensional scatter when compared to the

rank alternatives was higher than when compared with the scatter diagram. The variance associated with the rank alternatives was considerably higher than the other responses.

Specific comments stated the need for the axes to be identified, to show direction of improvement, and that the resolution made it difficult to interpret. It was noted that this display is a good idea, provides a good opportunity for identifying better alternatives, and has broader applications.

5.5.3 Advanced Three-Dimensional Scatter Plot

The SVT advanced three-dimensional scatter plot was compared to the LDW rank alternatives display and the LDW scatter diagram. The question dealt with how effective the displays were in representing the influence of each of the five criteria goals with respect to the overall CERCLA value.

5.5.3.1 Comparison 3-1

Comparison 3-1 was between the LDW rank alternatives and the SVT advanced three-dimensional scatter plot display. The paired t-Test identified the two means to be statistically equal. However, the three-dimensional scatter plot mean was 5.25 as compared to the rank alternative mean of 4.58. The associated p-value was 0.22.

5.5.3.2 Comparison 3-2

Comparison 3-2 was between the LDW scatter diagram and the SVT advanced three-dimensional scatter plot display. The paired t-Test identified the two means to be statistically equal. However, the three-dimensional scatter plot mean was 5.50 as compared to the rank alternative mean of 4.92. The associated p-value was 0.11.

5.5.3.3 Advanced Three-Dimensional Scatter Plot Assessment

The SVT advanced three-dimensional scatter plot display was shown to statistically have a mean equal to both the rank alternatives display and the scatter diagram and therefore statistically we could not reject the hypothesis that there is no difference between the value of the two graphs. It is interesting that the advanced three-dimensional scatter plot has every feature of the three-dimensional scatter plot, had higher means than the three-dimensional scatter plot, and was compared to the same LDW displays. Moreover, the means for both of the LDW displays improved in both comparisons when compared to the three-dimensional scatter plot comparisons.

One individual mentioned that it was a very good idea being able to size the cube based on a variable value; whereas, another commented they preferred a bar chart to a spatial visualization of data.

5.5.4 Goal Display

The SVT goal display was compared to the LDW rank alternatives display, the LDW stack bar ranking display, the LDW graph an alternative display, and the LDW compare alternatives graph. The question dealt with how effective the displays were in comparing the performance of the alternatives for selected goals.

5.5.4.1 Comparison 4-1

Comparison 4-1 was between the LDW rank alternatives and the SVT goal display. The paired t-Test identified the two means to be statistically equal. However,

the goal display mean was 5.08 as compared to the rank alternative mean of 4.83. The associated p-value was quite high at 0.62.

5.5.4.2 Comparison 4-2

Comparison 4-2 was between the LDW stack bar ranking and the SVT goal display. The paired t-Test identified the two means to be statistically equal. However, the goal display mean was 5.25 as compared to the rank alternative mean of 4.92. The associated p-value was 0.42.

5.5.4.3 Comparison 4-3

Comparison 4-3 was between the LDW graph an alternative and the SVT goal display. The paired t-Test identified the two means to be statistically equal. The goal display mean was 5.08 as compared to the rank alternative mean of 4.67. The associated p-value was 0.24.

5.5.4.4 Comparison 4-4

Comparison 4-4 was between the LDW compare alternatives graph and the SVT goal display. The paired t-Test identified the two means to be statistically equal. However, the goal display mean was 4.83 as compared to the rank alternative mean of 5.67. The associated p-value was very low at .054.

5.5.4.5 Goal Display Assessment

In all of the comparisons, the SVT goal display and all four of the LDW displays were shown to statistically have no differences in means. The means were very close in the first three comparisons; however, when the goal display was compared with the

compare alternatives graph, the p-value was extremely close to the alpha of 0.05. One potential area of concern was that the goal display does not have a "one specific" application and therefore did have an exact one-to-one mapping with a LDW display. However, the mean was above five in the first three comparisons and just below in the last.

Specific comments mentioned that the LDW displays were good, but limited; the SVT display is unlimited, but not as good because it doesn't reach its full potential. Another comment mentioned that all of the information provided by LDW should be incorporated into the SVT display. As mentioned in Section 5.2, this particular display was not very effective because the information panel was not available in the software version that was validated.

5.5.5 Sensitivity Analysis Goal Display

The SVT sensitivity analysis goal (SAG) display was compared to the LDW sensitivity graph, the LDW sensitivity table, and the LDW sensitivity diagrams. The question dealt with performing sensitivity analysis (changing the weight) on the five criteria goals and then representing the new information to the user.

5.5.5.1 Comparison 5-1

Comparison 5-1 was between the LDW dynamic sensitivity display and the SVT sensitivity analysis goal display. The paired t-Test identified the two means to be statistically different. The mean response for the dynamic sensitivity display was 4.42 and the mean response for the sensitivity analysis goal display was just over 5.58 with an associated p-value of 0.01.

5.5.5.2 Comparison 5-2

Comparison 5-2 was between the LDW sensitivity graph and the SVT sensitivity analysis goal display. The paired t-Test identified the two means to not be statistically different. The mean response for the dynamic sensitivity display was 5.17 and the mean response for the sensitivity analysis goal display was approximately 5.67 with an associated p-value of 0.19.

5.5.5.3 Comparison 5-3

Comparison 5-3 was between the LDW sensitivity table and the SVT sensitivity analysis goal display. The paired t-Test identified the two means to be statistically different. The mean response for the dynamic sensitivity display was 4.33 and the mean response for the sensitivity analysis goal display was again approximately 5.67 with an associated p-value of 0.0012.

5.5.5.4 Sensitivity Analysis Goal Display Assessment

In all three of the comparisons, the SVT sensitivity analysis goal display had a higher mean than any of the three LDW displays. However, statistically it was determined that there was only a difference between the sensitivity analysis goal display and the LDW dynamic sensitivity display and the sensitivity table. This particular group of comparisons had direct mappings of functional capabilities.

Specific comments mentioned that the SVT sensitivity analysis goal display should automate the changing of weights. Currently, there is a scrollbar to adjust the weights to the desired values. Another comment stated that a continuous three-dimensional surface might be better rather than choosing independent values. It was also noted that specific

values should be placed on the rectangular cubes as well as mark the values on the surface of the cube.

5.5.6 Measure Display

The SVT measure display was compared to the LDW rank alternatives display, the LDW stack bar ranking display, the LDW graph an alternative display, and the LDW compare alternatives graph. The question dealt with how effective the displays were in comparing the performance of the alternatives for selected measures.

5.5.6.1 Comparison 6-1

Comparison 6-1 was between the LDW rank alternatives and the SVT measure display. The paired t-Test identified the two means to be statistically equal. In fact, the means for both displays were 5.00 and the associated p-value was extremely high at 1.00.

5.5.6.2 Comparison 6-2

Comparison 6-2 was between the LDW rank alternatives and the SVT measure display. The paired t-Test identified the two means to be statistically equal. However, the measure display mean was 5.17 as compared to the rank alternative mean of 4.58. The associated p-value was 0.19.

5.5.6.3 Comparison 6-3

Comparison 6-3 was between the LDW rank alternatives and the SVT measure display. The paired t-Test identified the two means to be statistically equal. However, the goal display mean was 5.25 as compared to the rank alternative mean of 5.75. The associated p-value was quite low at 0.08.

5.5.6.4 Comparison 6-4

Comparison 6-4 was between the LDW rank alternatives and the SVT measure display. The paired t-Test identified the two means to be statistically equal. However, the goal display mean was 5.17 as compared to the rank alternative mean of 5.33. The associated p-value was quite high at 0.66.

5.5.6.5 Measure Display Assessment

In all of the comparisons, the SVT goal display and all four of the LDW displays were shown to statistically have no differences in means. The means were very close in the all of the comparisons; however, when the measure display was compared with the graph an alternative, the p-value was extremely close to the alpha of 0.05. One potential area of concern was that like the goal display, the measure display does not have "one specific" application and therefore did not have an exact one-to-one mapping with a LDW display. However, the mean for all of the SVT displays was at least five or higher in all four comparisons.

Specific comments mentioned that viewing all of the data at one time is a bit overwhelming and that labeling of the information was essential.

5.5.7 Animated Alternatives

The SVT animated alternative display was compared to the LDW graph an alternative display and the LDW compare alternatives graph. The question dealt with how effective the displays were in comparing two alternatives based upon their respective measure scores.

5.5.7.1 Comparison 7-1

Comparison 7-1 was between the LDW graph an alternative and the SVT animated alternative display. The paired t-Test identified the two means to be statistically equal. The animated alternative display mean was 5.17 as compared to the graph an alternative display mean of 5.08. The associated p-value was extremely high at 0.86.

5.5.7.2 Comparison 7-2

Comparison 7-2 was between the LDW compare alternatives graph and the SVT animated alternative display. The paired t-Test identified the two means to be statistically equal. The animated alternative display mean was 5.08 as compared to the compare alternatives graph mean of 5.50. The associated p-value was 0.45.

5.5.7.3 Animated Alternatives Assessment

In both of the comparisons, the SVT animated alternatives display and the two LDW displays had similar means. Statistically, it was determined that there was no difference between the animated alternative display and the LDW graph an alternative display and compare alternatives graph.

It was noted that providing numbers would help in using this display. Also, the proper labeling of the selected objects was crucial for understanding. Another comment suggested displaying more alternatives than just one or two. This display, as well as the animated measures display (below), have both been improved. When one object is selected, the corresponding object is automatically selected as well. This eliminates any

confusion as to which objects are matched pairs. Also, the information screen has been fully implemented in all displays.

5.5.8 Animated Measures

The SVT animated measure display was compared to the LDW scatter diagram. The question dealt with how effective the displays were in comparing two measures based upon the values of all 23 alternatives in those two measures.

5.5.8.1 Comparison 8-1

Comparison 8-1 was between the LDW scatter diagram and the SVT animated measure display. The paired t-Test identified the two means to be statistically equal. In fact, they both had a mean of 5.17 and the associated p-value was 1.00.

5.5.8.2 Animated Measures Assessment

The SVT animated measures display and the LDW scatter diagram display had similar means. Statistically, it was determined that there was no difference between the animated measure display and the LDW scatter diagram. Because some much needed functionality (as discussed in animated alternatives above) was not available during the validation process, the effectiveness of this display was diminished.

One comment mentioned this might be better served as a stacked bar chart. Also, the need for labeling the information is crucial.

5.5.9 Overall

The final question was concerned with the amount of user interaction provided with the software displays (being able to control certain features such as viewpoint,

number of variables displayed, color, etc) and whether the amount of user interaction helped increase overall understanding of the data set and any analysis performed.

5.5.9.1 Comparison Overall

The final comparison was between the overall user interaction of both software programs, LDW and SVT. The paired t-Test identified the two means to be statistically equal. The mean for LDW was 5.25 and the mean for the SVT was 5.58. The associated p-value was 0.44.

5.5.9.2 Overall Assessment

The means were fairly close in this comparison. However, some valuable written comments were provided. One individual stated the software visualization tool had huge potential, not just in display but in a virtual comparison environment. Another individual mentioned that with respect to visualization the package is great. They really liked the ability to zoom and rotate. One other person liked the tool a lot, but felt it is not yet a replacement for LDW. A couple of individuals felt that using three-dimensions made it more difficult to see what was happening, that it actually was more harmful than helpful. Lastly, the need for labeling of the information was mentioned throughout the surveys.

5.6 Overall Analysis

Although the software visualization tool was not fully implemented, the scores that the tool received were overall favorable. In most cases the software visualization tool displays had a higher mean than the corresponding LDW displays; however, it was only statistically higher with respect to the weight hierarchy display, the scatter plot display,

and the sensitivity analysis goal display. Not having the information screen available hindered the effectiveness of all the displays, but had a significantly negative impact on the goal and measure displays. The biggest problems with the software visualization tool have already been addressed and corrections implemented, specifically the information screen and the highlighting of the corresponding cube or sphere in both the animated alternative and animated measure displays.

5.7 Conclusions

Although the validation process was not as successful as originally planned, it did provide some beneficial insights into the potential of the software visualization tool. Certain displays, even without the information screen implemented, demonstrated their potential to help improve the understanding and insights gained from an operations research analysis. Specifically, the weight hierarchy, scatter plots, and the sensitivity analysis goal display appeared to have the greatest potential. The software developed can now serve as a prototype for what could eventually become a fully implemented software application. It has become apparent that additional validation, involving a variety of analysts and decisions makers, should be conducted in order to further define the requirements of the tool from both the analysts and decision maker's perspective.

6. Conclusions and Recommendations

This chapter provides conclusions from the analysis and makes recommendations for further study of the topic area.

6.1 Overview

One of the major tasks in operations research is analyzing data and obtaining the results. Effective visualization tools are needed for exploring data sets and locating regions of interest, often interactively and in real time. Visualization offers a powerful tool for communicating meaning to others. Powerful desktop computers and appropriate visualization software enable a user to analyze data in a powerful and flexible way. Having data is one thing; extracting real and meaningful information from it is something quite different, but most importantly is then being able to conduct an accurate analysis of the information. The power of visual exploration and presentation of the data allow the user to look at the data and to study features and areas of interest previously unforeseen. A visualization that clearly conveys the meaning of data not only helps explain the researcher's idea, but also enhances credibility and capability. Information visualization, especially when dealing with multivariate and higher dimensional data, is an emerging technology.

This thesis effort accomplished the goal of examining methods to visualize multidimensional data. The methods presented in this thesis are by no means complete or comprehensive. However, they do demonstrate the potential added value that visualization of data can have upon an operations research analysis in general and

specifically a multiple attribute preference theory analysis. Selecting a specific approach to view the multidimensional data is a very complicated process. Each analysis is unique as are the visualizations required to best represent the associated data set.

6.2 Limitations of the Study

Although the software visualization tool only served as a prototype, it did produce some graphical displays that provided better insight into the data set. The limitations were due largely in part to a lack of computer graphics and visual programming knowledge and experience. Also, limited availability of programming tools (integrated editor, compiler, debugger) and programming software (using an evaluation version 3D API) hindered the actual software implementation.

6.3 Recommendations and Future Research

The potential of using visualization to better understand and comprehend the data was clearly demonstrated. Section 3.6, Promising Approaches, discussed several additional areas of interes with potential for creating a more complete software visualization tool. Of particular interest is the field of virtual reality. A follow-on thesis by a student in either a computer science or software engineering field of study could be conducted to create a fully operational software visualization tool, possibly incorporating virtual reality.

6.4 Conclusion

The goal of using multivariate visualization techniques to display the data associated with an operations research analysis was to improve upon the understanding

and insights into the data set being studied. The software visualization tool incorporated several multivariate techniques, such as three-dimensions (3D), glyphs, parallel axes, and animation, in order to effectively display higher dimensional data. The software visualization tool created several displays to visually represent the multivariate data.

Three displays in particular, the weight hierarchy display, the three-dimensional scatter plot display, and the sensitivity analysis goal display were identified during the validation process for demonstrating the greatest potential to provide additional insight and understanding into an operations research analysis. The weight hierarchy display clearly represented the overall structure of the data and identified the relative contribution of each goal and measure to the overall CERCLA value. The three-dimensional scatter plot display was able to provide more information than the two-dimensional plot and with less clutter. Most noteworthy was the sensitivity analysis goal display's ability to perform sensitivity analysis on the criteria goals' weight and the CERCLA value and then immediately display the results. Also, the ability to navigate through the data by zooming in and out of the display and/or rotating the view itself provided the user with multiple views and helped to increase the overall understanding of the data set being analyzed.

Clearly, the field of information visualization offers tremendous benefits. However, there is not one single display or visualization technique that can be applied to best represent a data set. The challenge lies in finding or creating the specific visualization software tool that most accurately displays the data and provides the greatest insight into the data set being studied.

Appendix A: LDW and SVT Visualization Displays

The following sections show the Logical Decisions for Windows (LDW) and the software visualization tool (SVT) displays used in the comparisons.

Comparison 1-1: LDW Bubble Diagram and SVT Weight Hierarchy

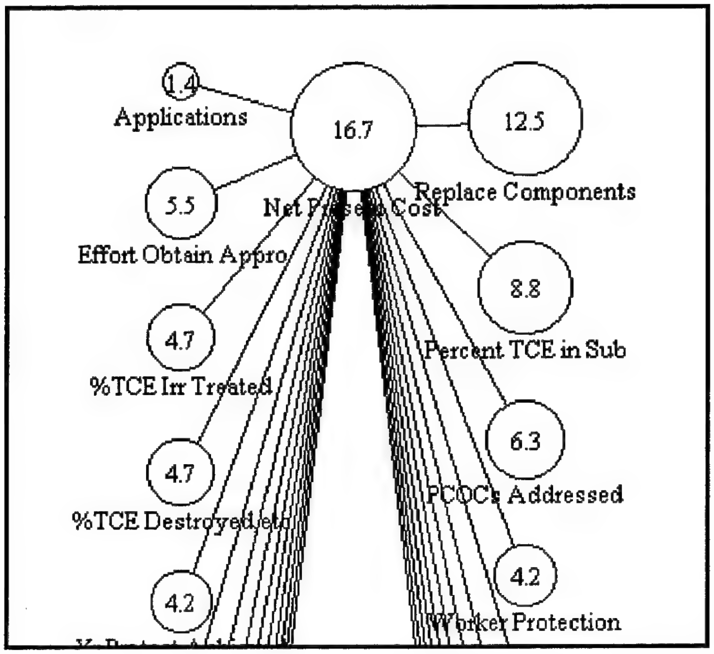


Figure 13. LDW Bubble Diagram

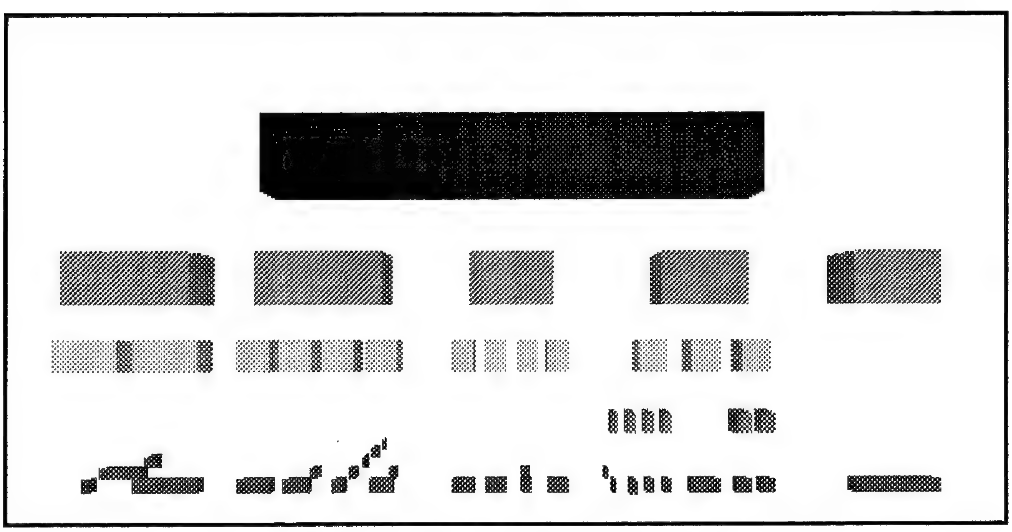


Figure 14. SVT Weight Hierarchy

Comparison 1-2: LDW Goals Hierarchy and SVT Weight Hierarchy

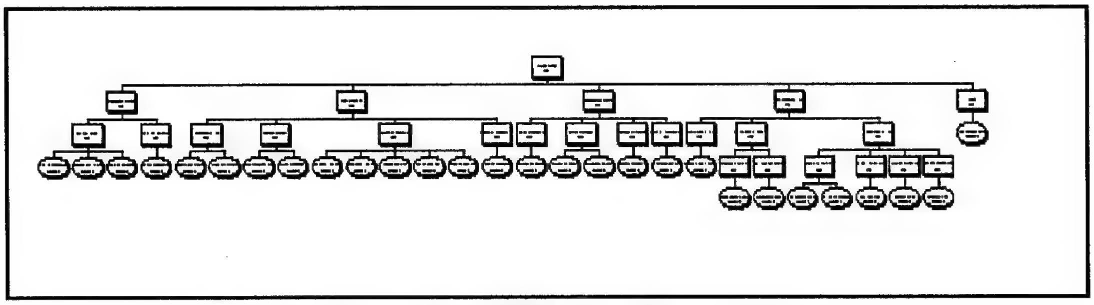


Figure 15. LDW Goals Hierarchy

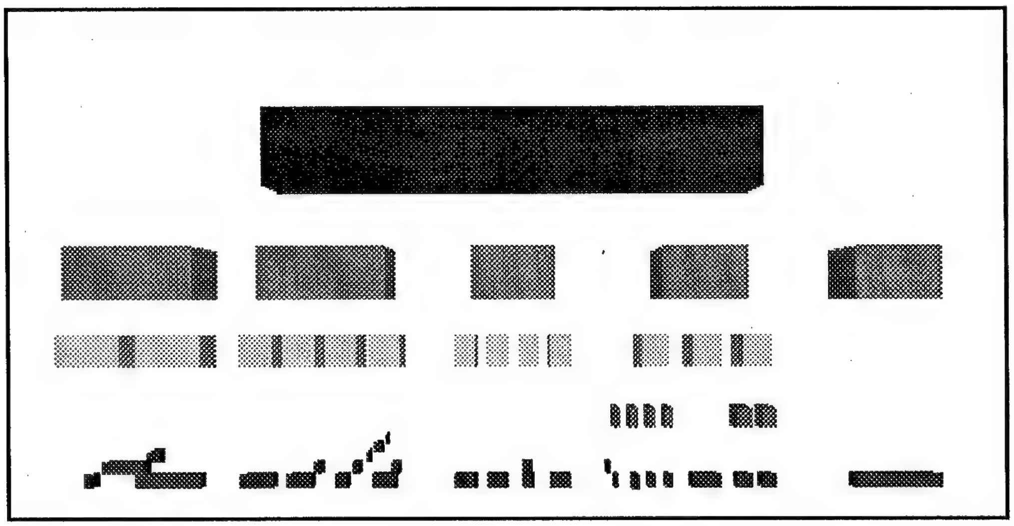


Figure 16. SVT Weight Hierarchy

Comparison 2-1: LDW Rank Alternatives and SVT 3-D Scatter Plot

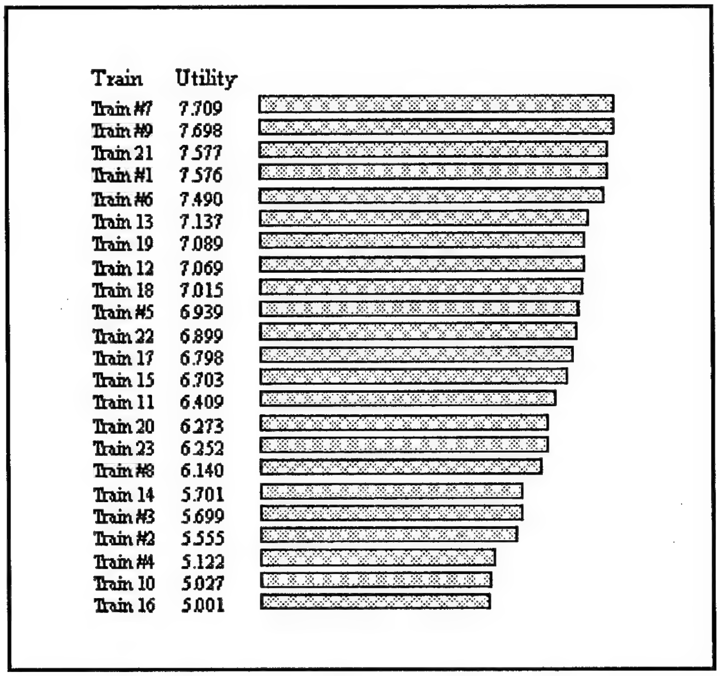


Figure 17. LDW Rank Alternatives

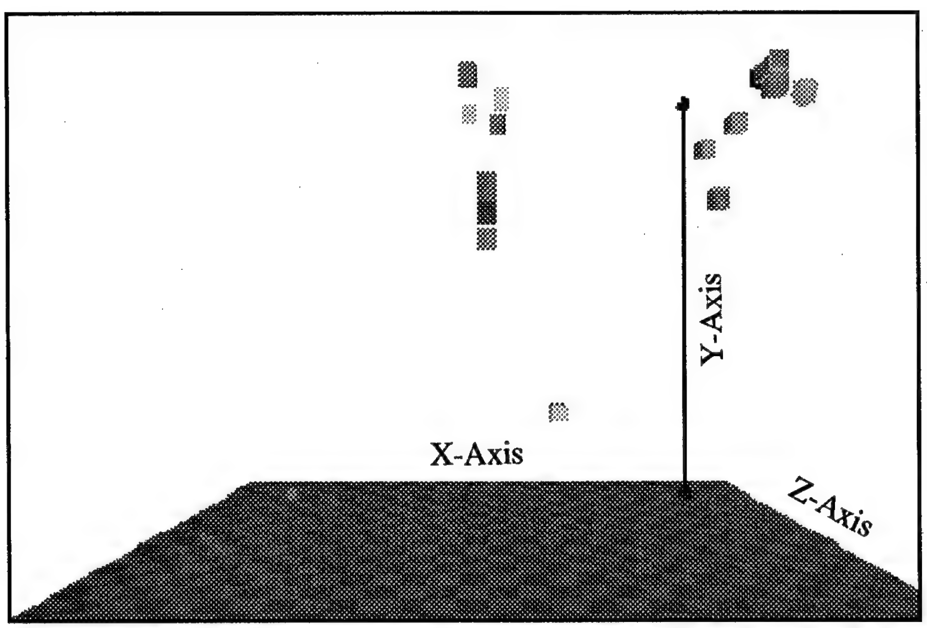


Figure 18. SVT 3-D Scatter Plot

Comparison 2-2: LDW Scatter Diagram and SVT 3-D Scatter Plot

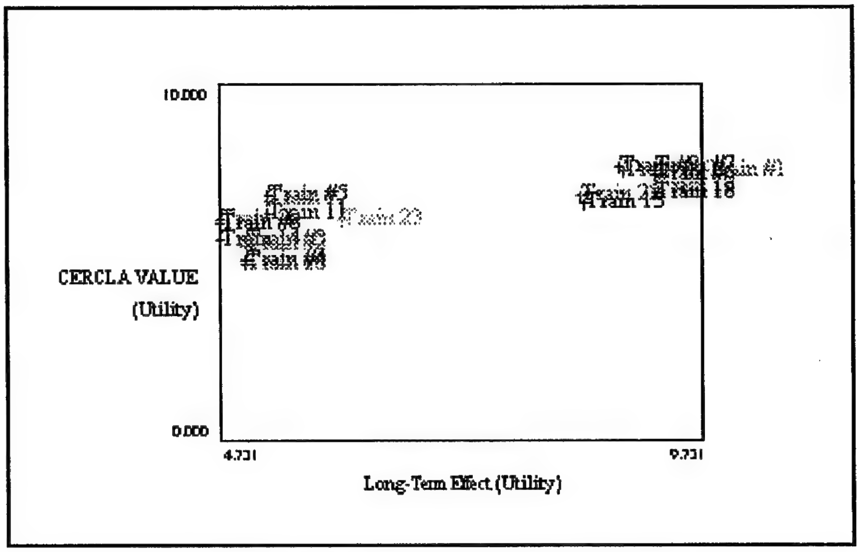


Figure 19. LDW Scatter Diagram

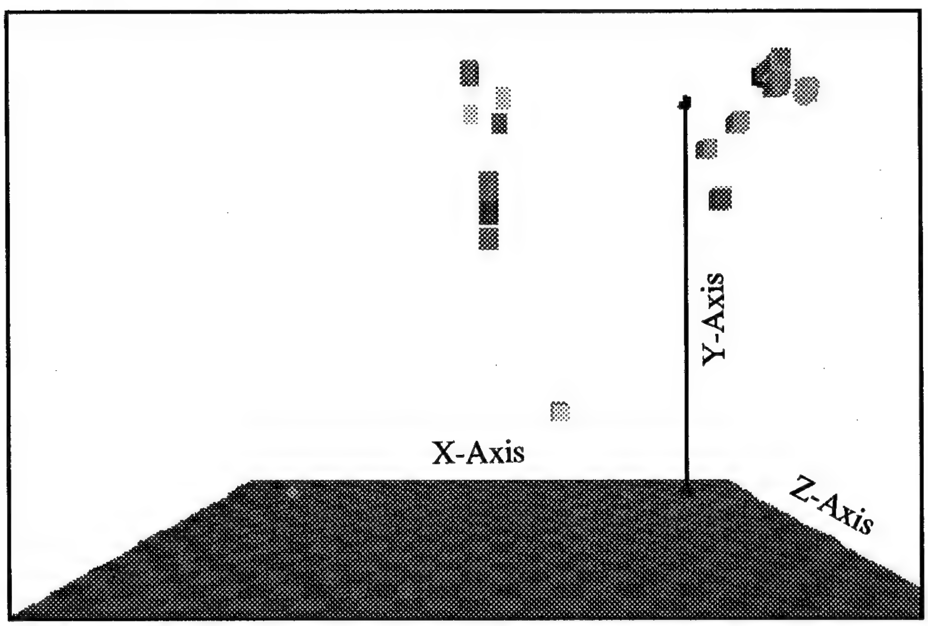


Figure 20. SVT 3-D Scatter Plot

Comparison 3-1: LDW Rank Alternatives and SVT Advanced 3-D Scatter Plot

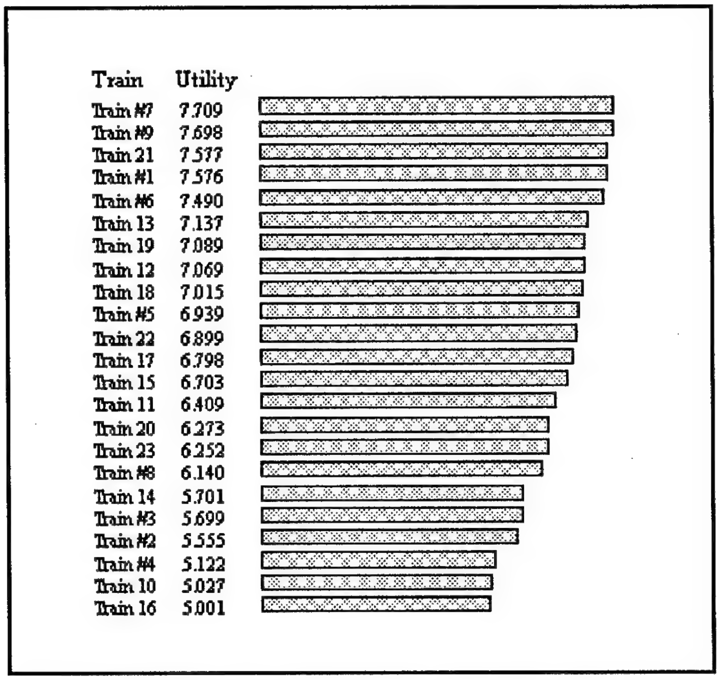


Figure 21. LDW Rank Alternatives

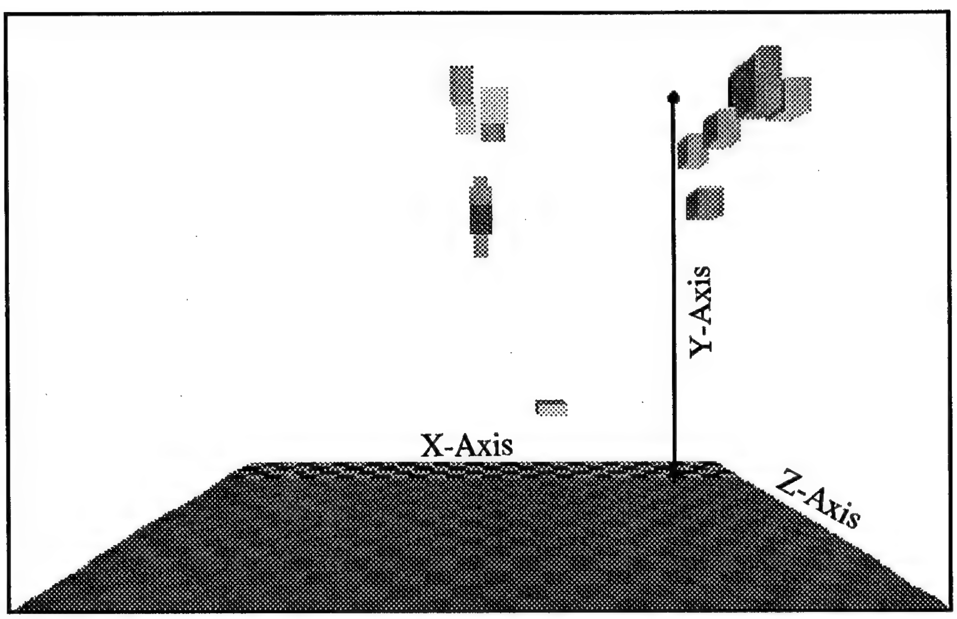


Figure 22. SVT Advanced 3-D Scatter Plot

Comparison 3-2: LDW Scatter Diagram and SVT Advanced 3-D Scatter Plot

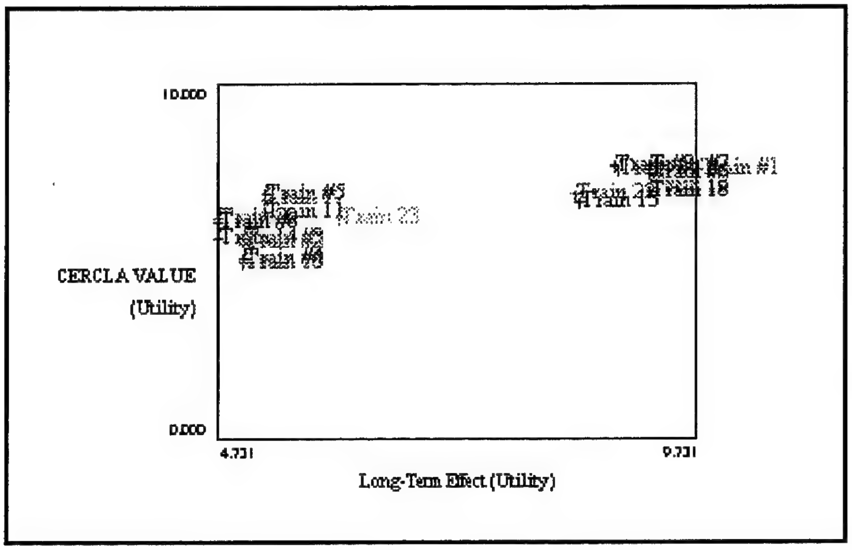


Figure 23. LDW Scatter Diagram

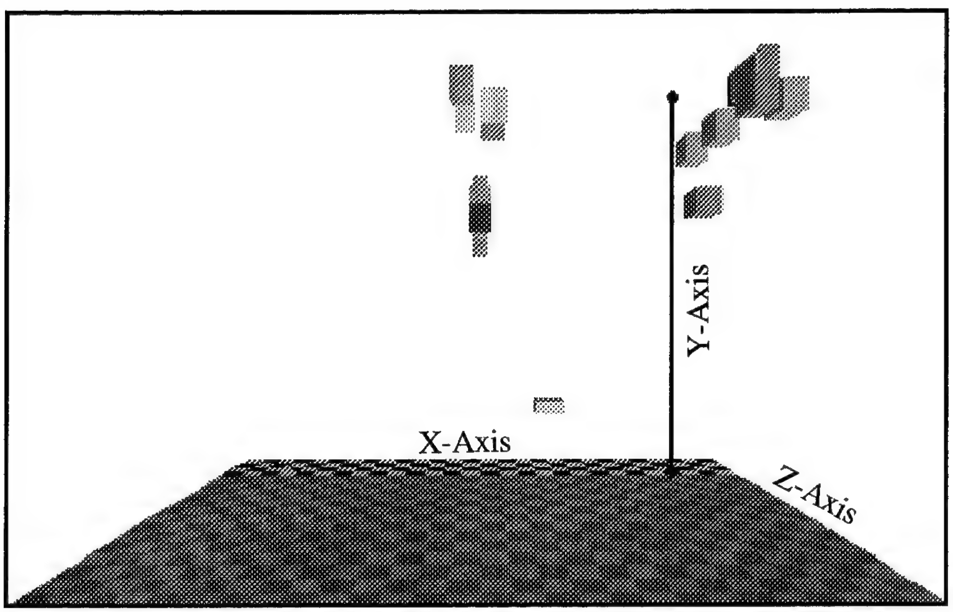


Figure 24. SVT Advanced 3-D Scatter Plot

Comparison 4-1: LDW Rank Alternatives and SVT Goal Display

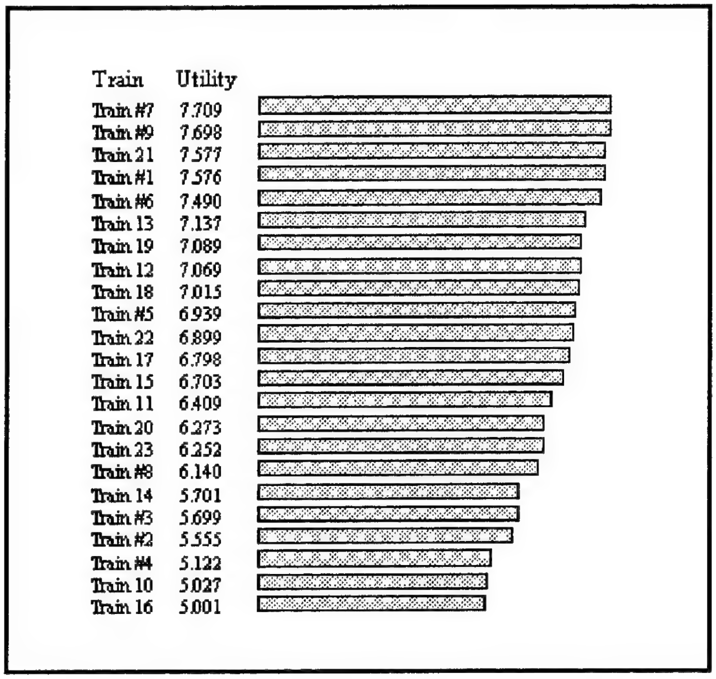


Figure 25. LDW Rank Alternatives

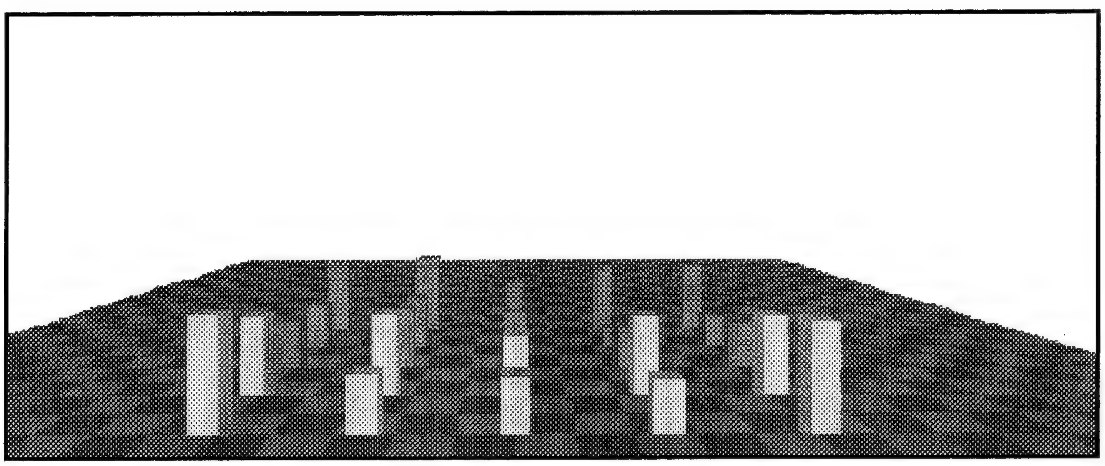


Figure 26. SVT Goal Display

Comparison 4-2: LDW Stack Bar Ranking and SVT Goal Display

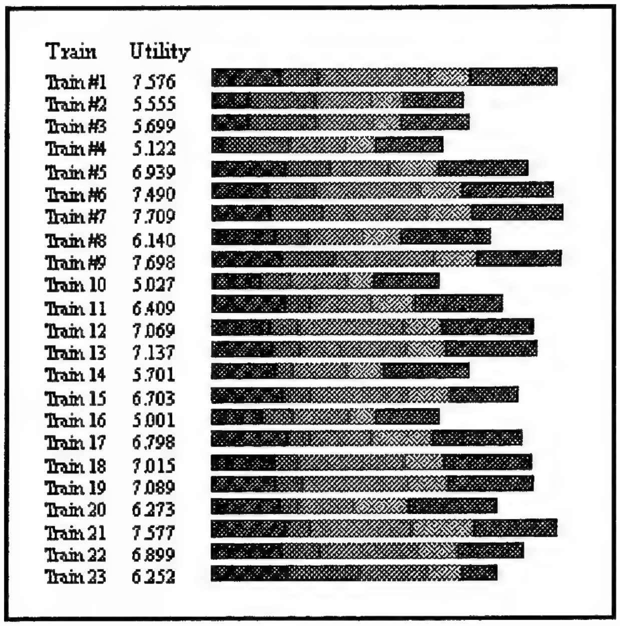


Figure 27. LDW Stack Bar Ranking

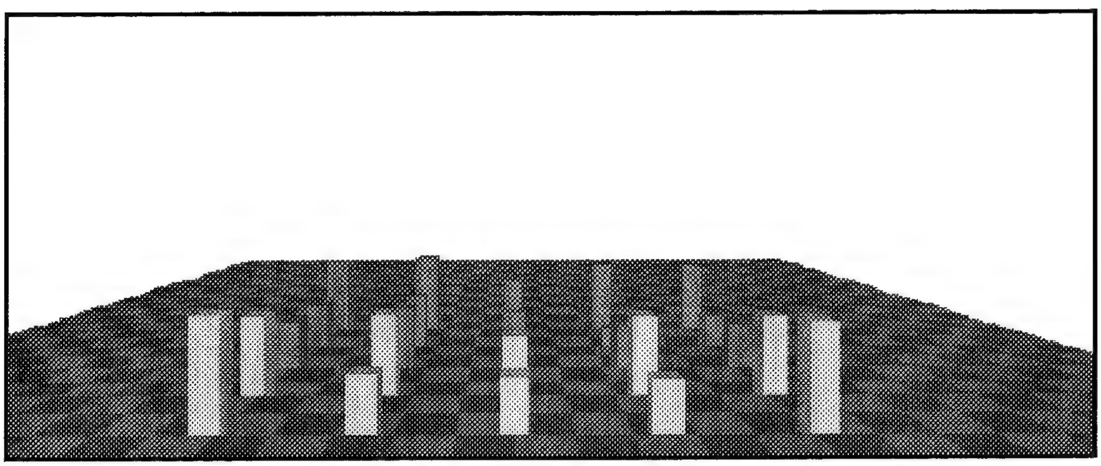


Figure 28. SVT Goal Display

Comparison 4-3: LDW Graph an Alternative and SVT Goal Display

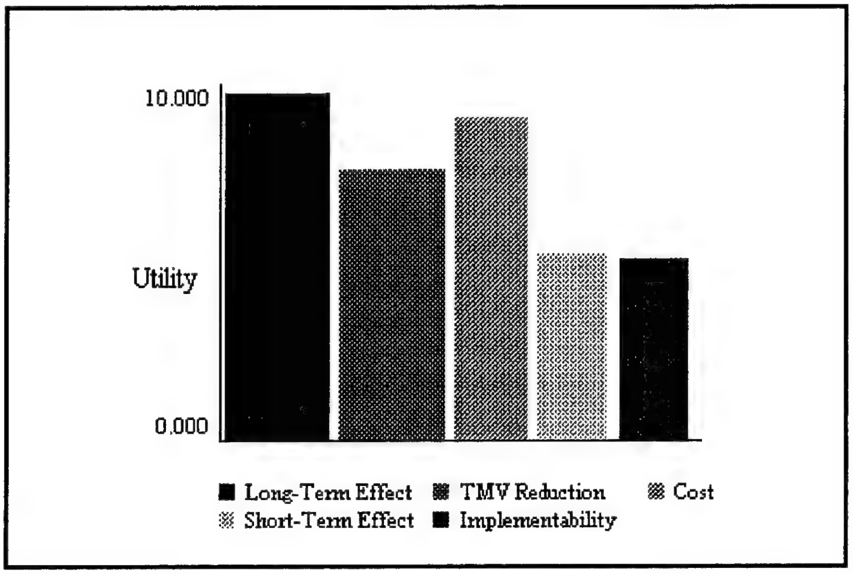


Figure 29. LDW Graph an Alternative

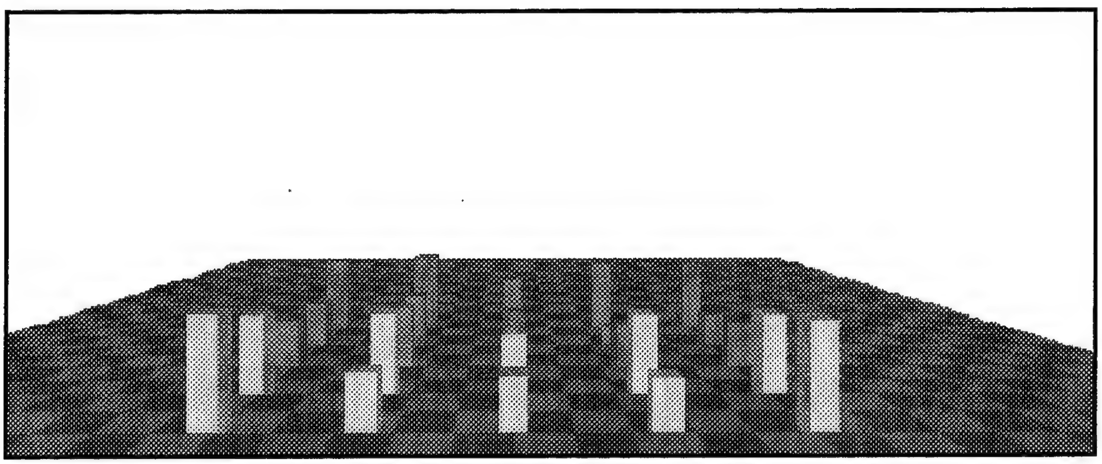


Figure 30. SVT Goal Display

Comparison 4-4: LDW Compare Alternatives Graph and SVT Goal Display

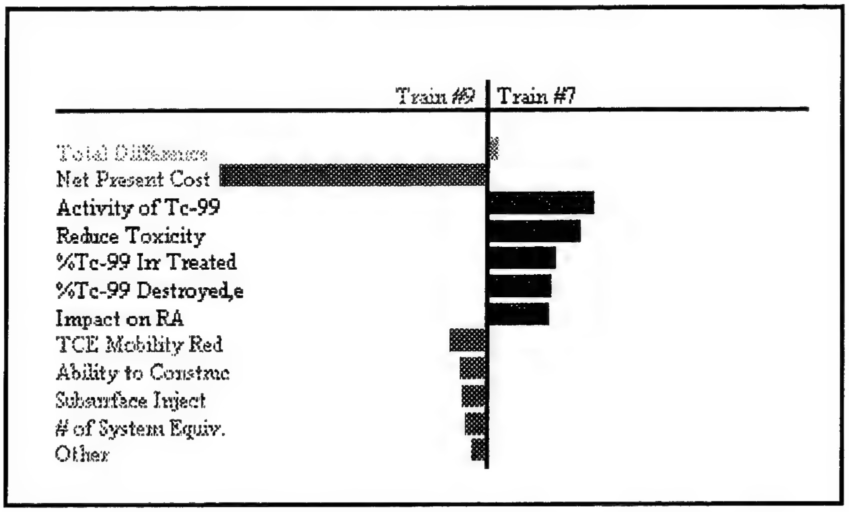


Figure 31. LDW Compare Alternatives Graph

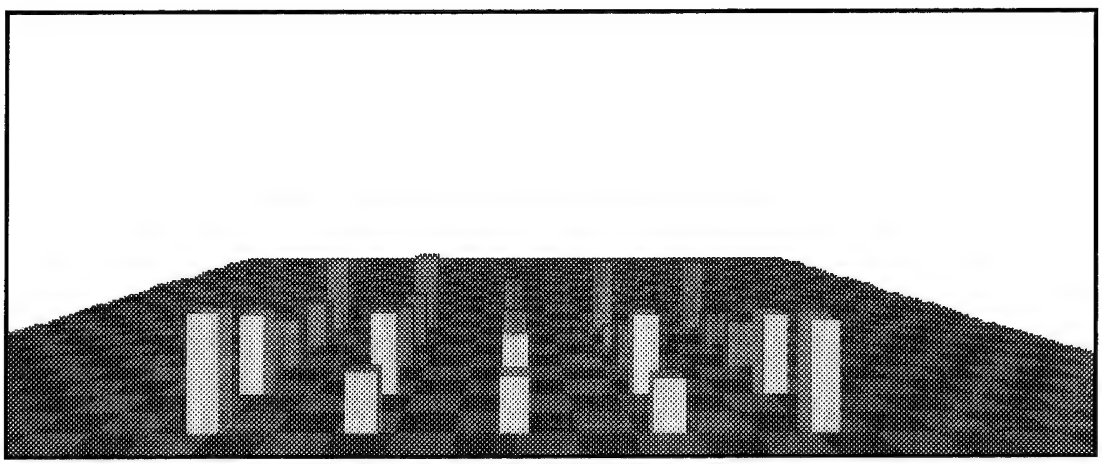


Figure 32. SVT Goal Display

Comparison 5-1: LDW Dynamic Sensitivity and SVT Sensitivity Analysis Goal Display

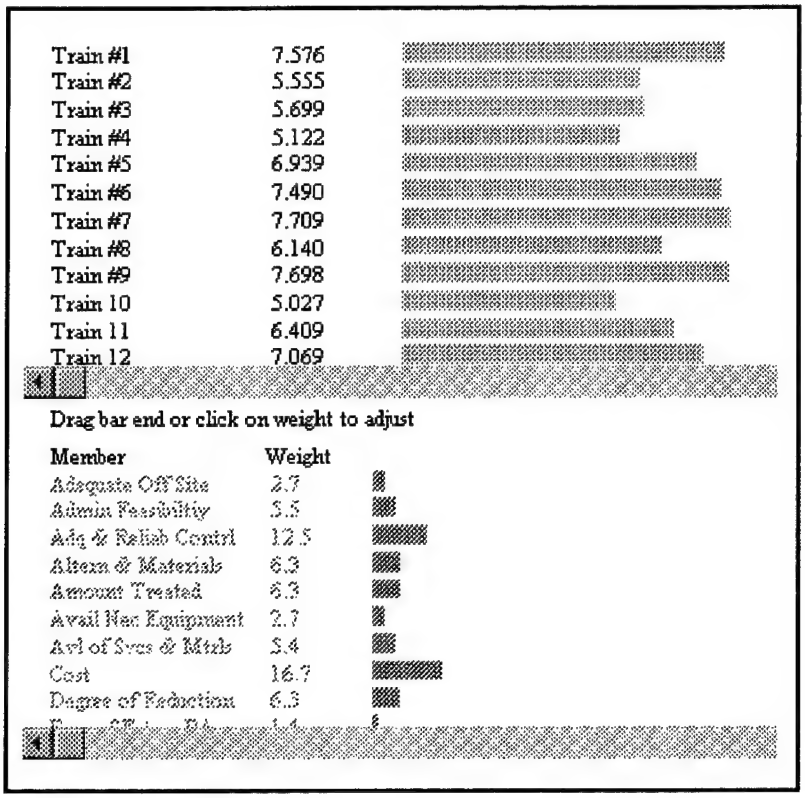


Figure 33. LDW Dynamic Sensitivity

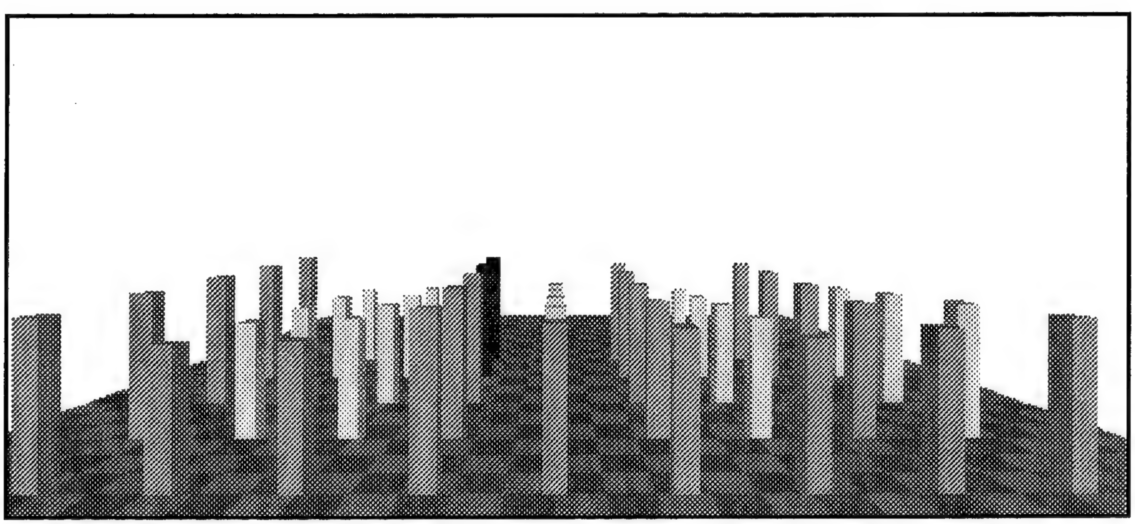


Figure 34. SVT Sensitivity Analysis Goal Display

Comparison 5-2: LDW Sensitivity Graph and SVT Sensitivity Analysis Goal Display

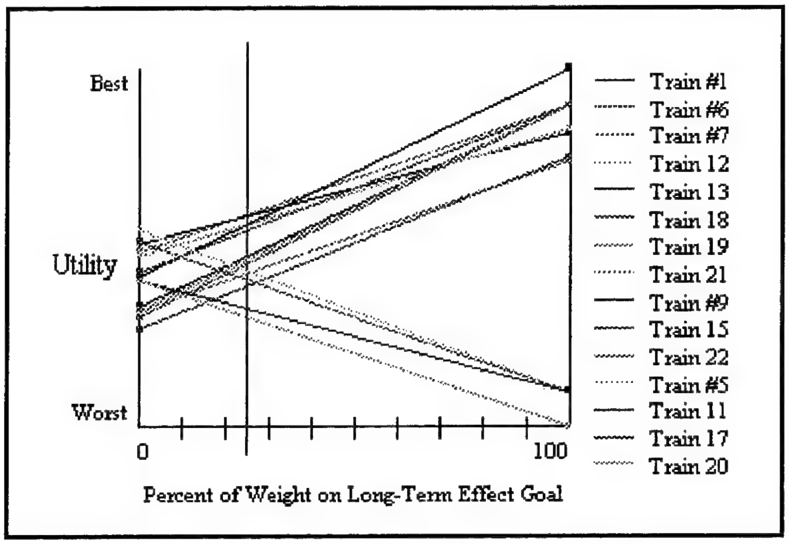


Figure 35. LDW Sensitivity Graph

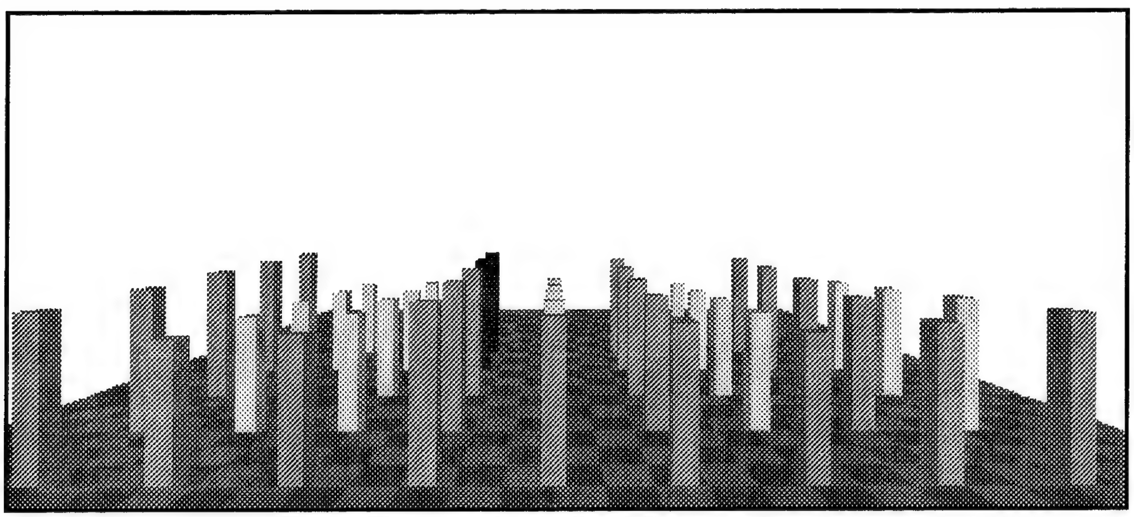


Figure 36. SVT Sensitivity Analysis Goal Display

Comparison 5-3: LDW Sensitivity Table and SVT Sensitivity Analysis Goal Display

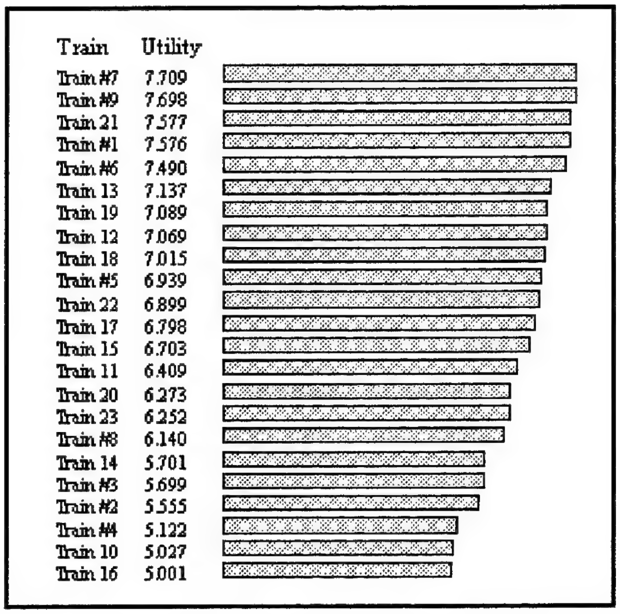


Figure 37. LDW Sensitivity Table

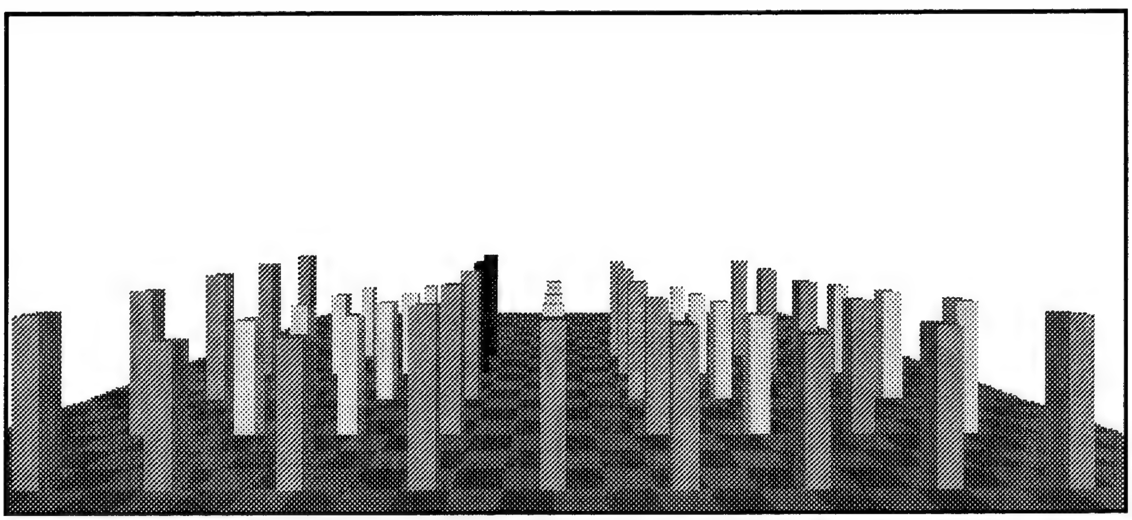


Figure 38. SVT Sensitivity Analysis Goal Display

Comparison 6-1: LDW Rank Alternatives and SVT Measure Display

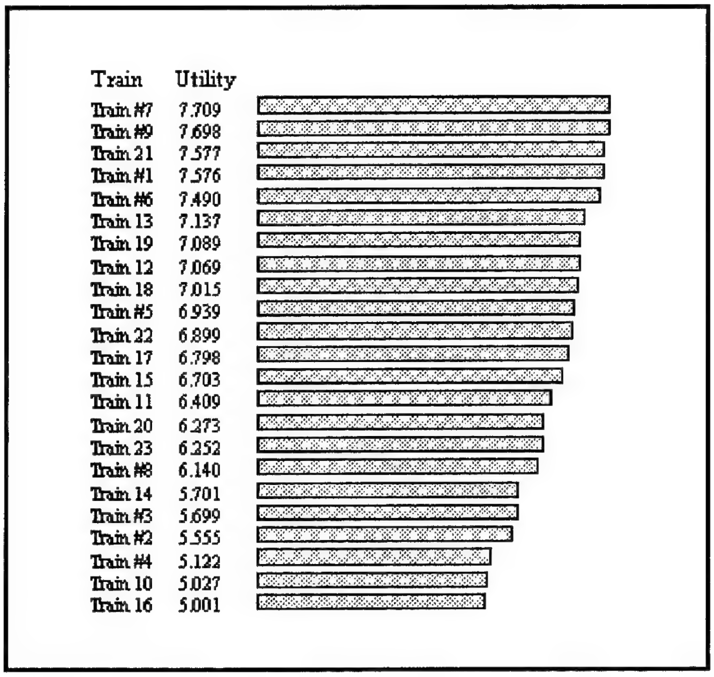


Figure 39. LDW Rank Alternatives

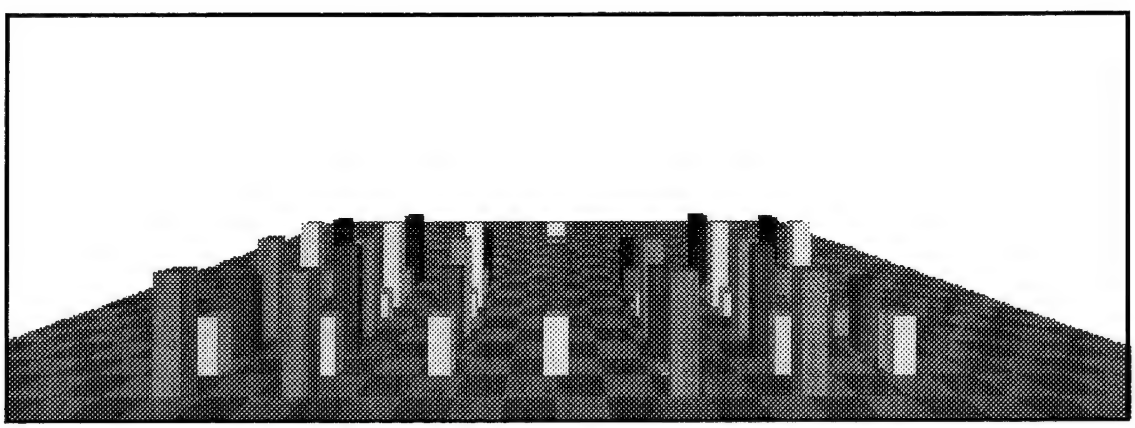


Figure 40. SVT Measure Display

Comparison 6-2: LDW Stack Bar Ranking and SVT Measure Display

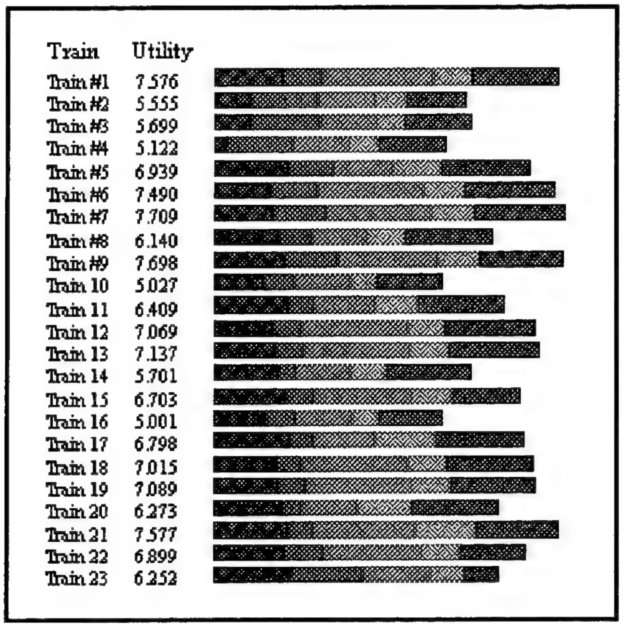


Figure 41. LDW Stack Bar Ranking

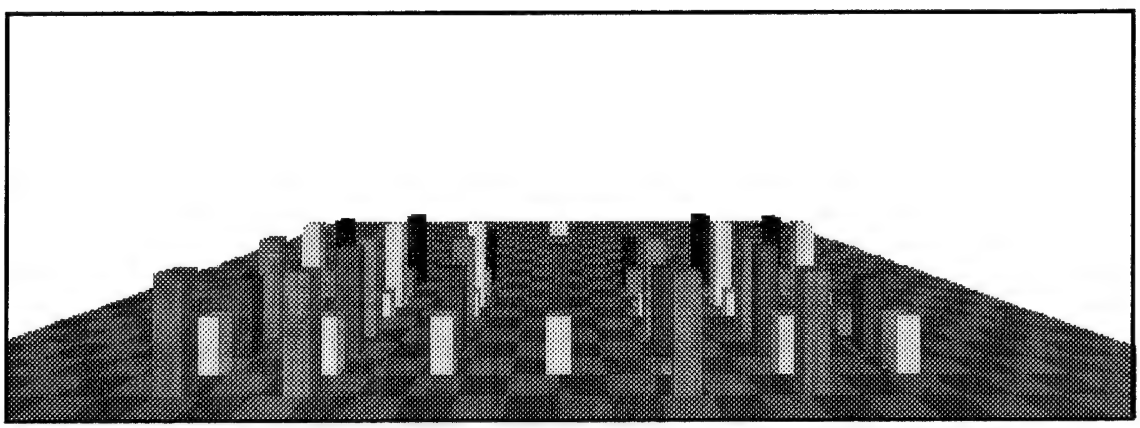


Figure 42. SVT Measure Display

Comparison 6-3: LDW Graph an Alternative and SVT Measure Display

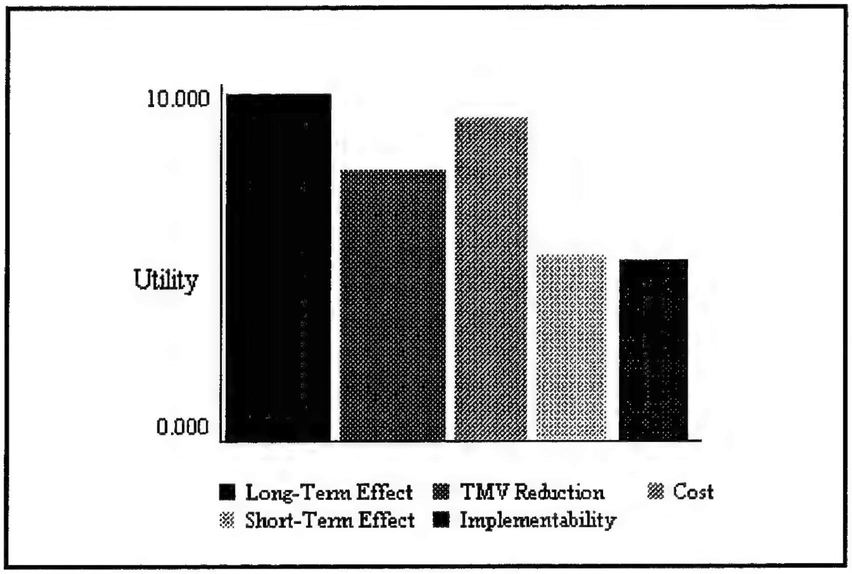


Figure 43. LDW Graph an Alternative

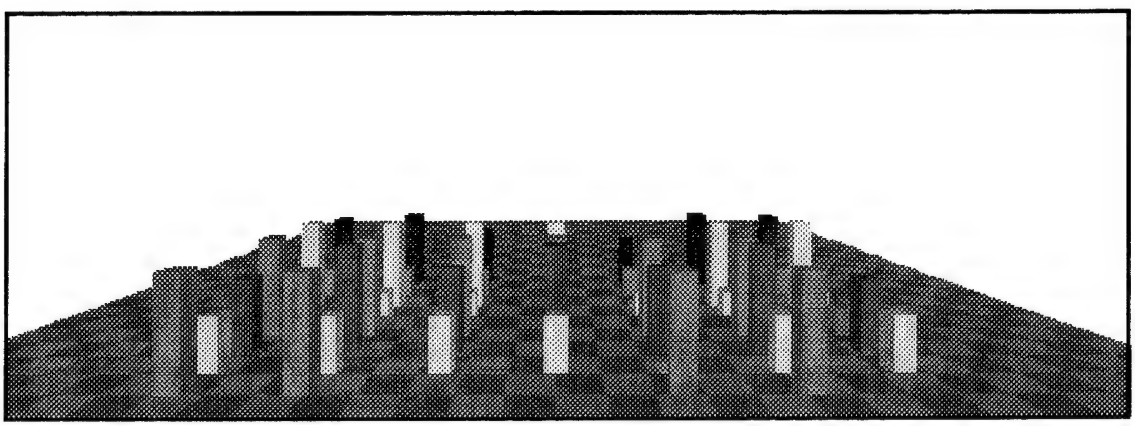


Figure 44. SVT Measure Display

Comparison 6-4: LDW Compare Alternatives Graph and SVT Measure Display

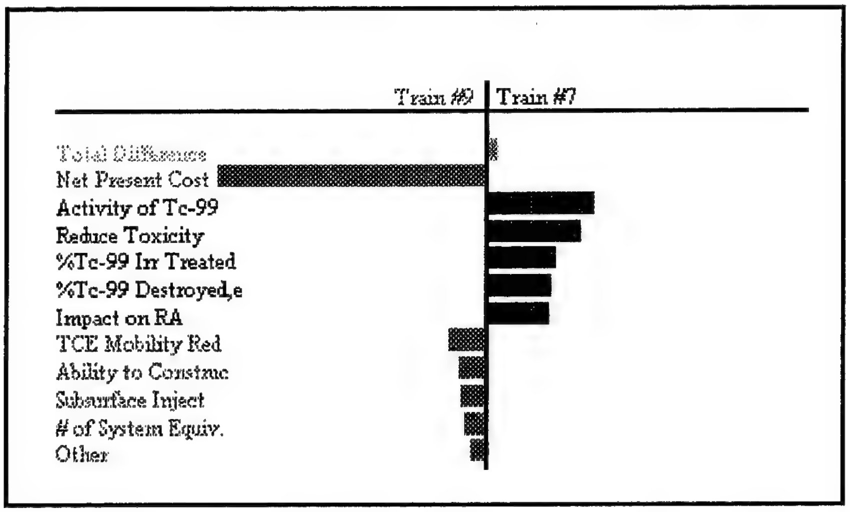


Figure 45. LDW Compare Alternatives Graph

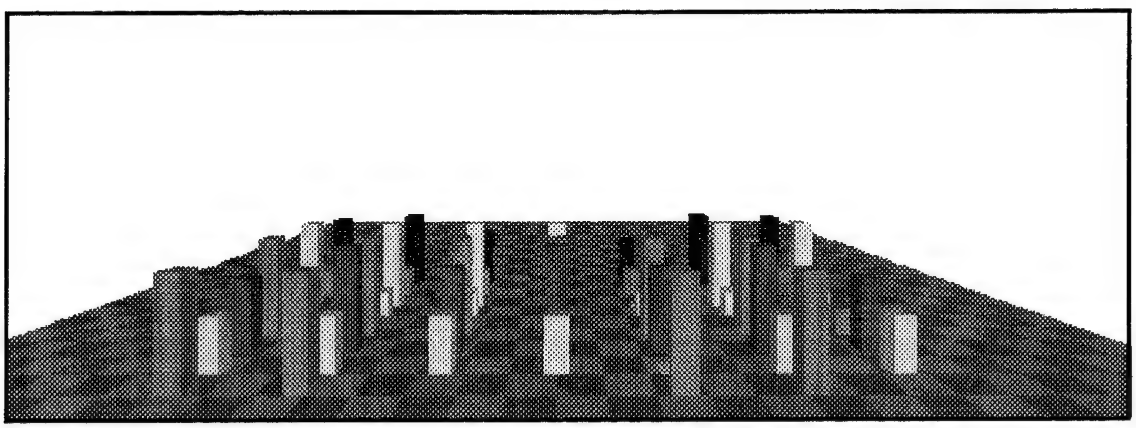


Figure 46. SVT Measure Display

Comparison 7-1: LDW Graph an Alternative and SVT Animated Alternative Display

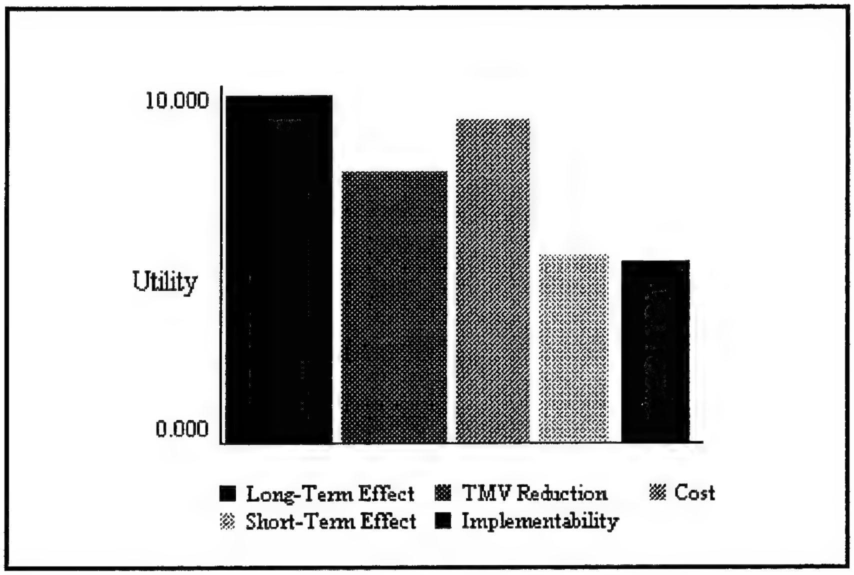


Figure 47. LDW Graph an Alternative

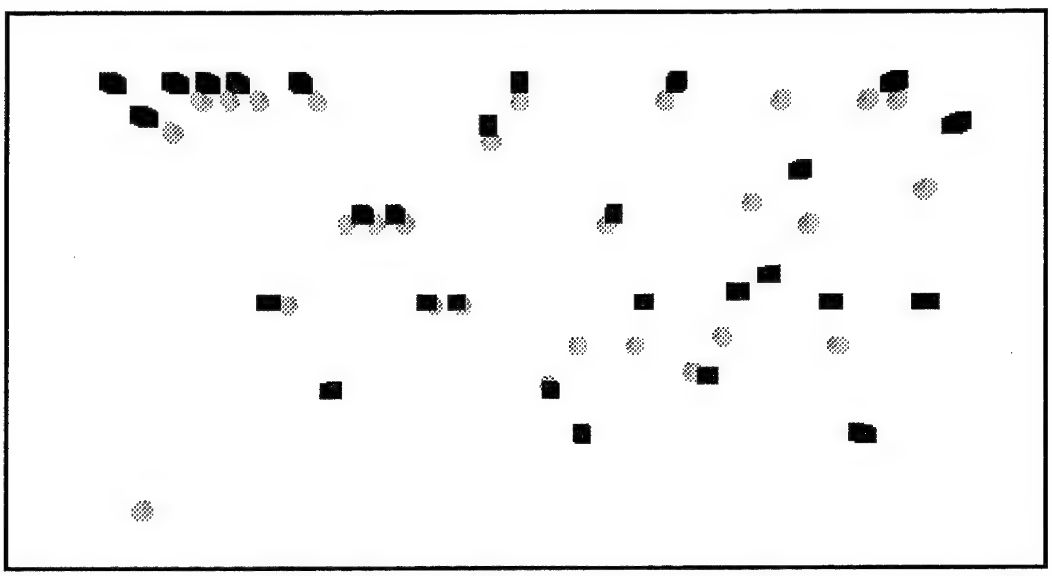


Figure 48. SVT Animated Alternative Display

Comparison 7-2: LDW Compare Alternatives Graph and SVT Animated Alternative Display

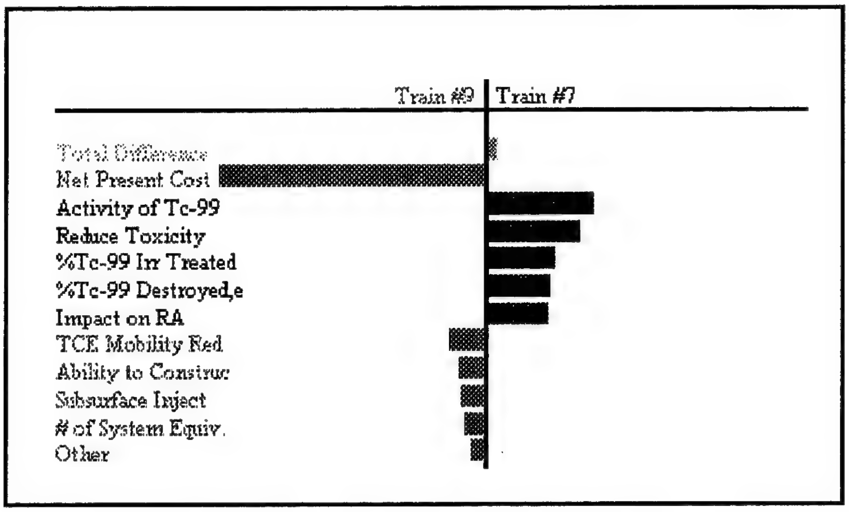


Figure 49. LDW Compare Alternatives Graph

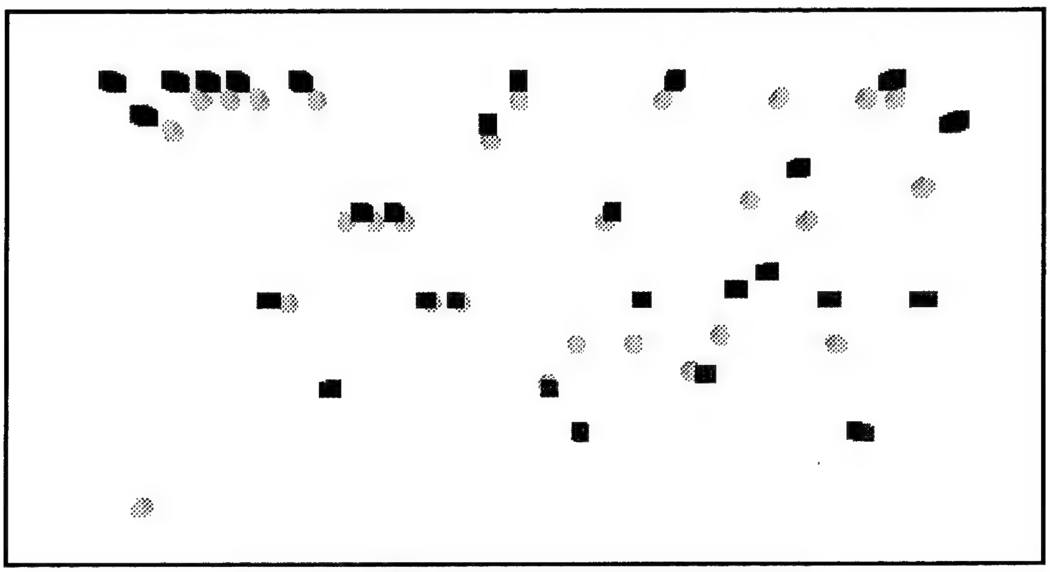


Figure 50. SVT Animated Alternative Display

Comparison 8-1: LDW Scatter Diagram and SVT Animated Measure

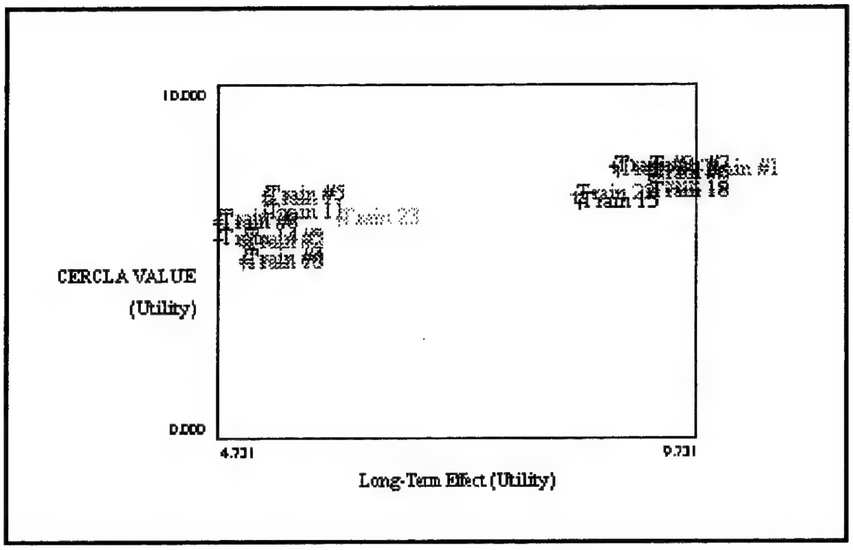


Figure 51. LDW Scatter Diagram

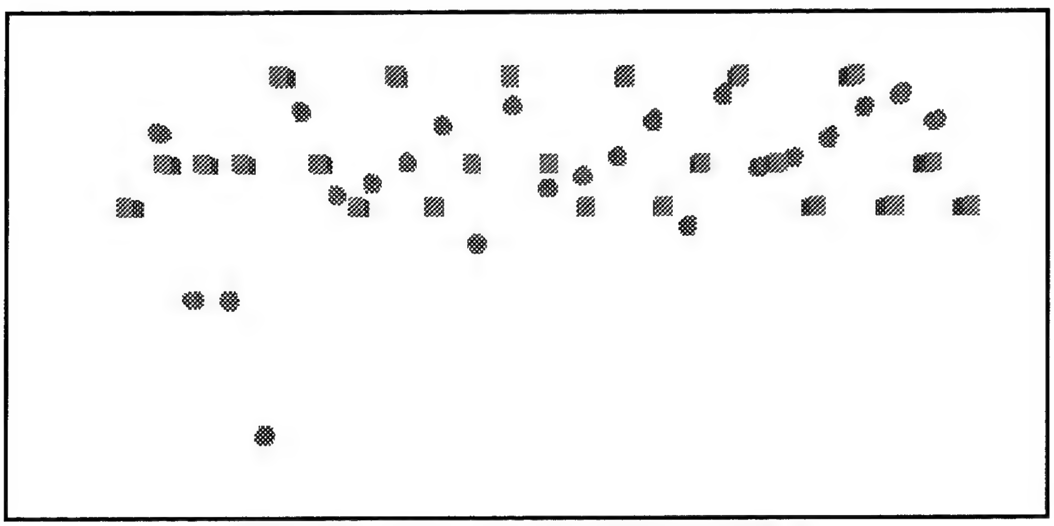


Figure 52. SVT Animated Measure Display

Appendix B: Test Results

For each comparison, a table is shown listing the means and variances for each software display. Also, the t statistic value, the p-value, and the t critical value are given. Two histograms are shown for each comparison, plotting the frequency of the actual responses.

Comparison 1-1: Bubble Diagram and Weight Hierarchy

Table 7. Comparison 1-1: Paired t-Test

Comparison 1-1	Bubble Diagram	Weight Hierarchy
Mean	4.67	5.42
Variance	0.61	1.36
t Stat	-2.28	REJECT
p-value	0.04	
t Critical two-tail	2.20	

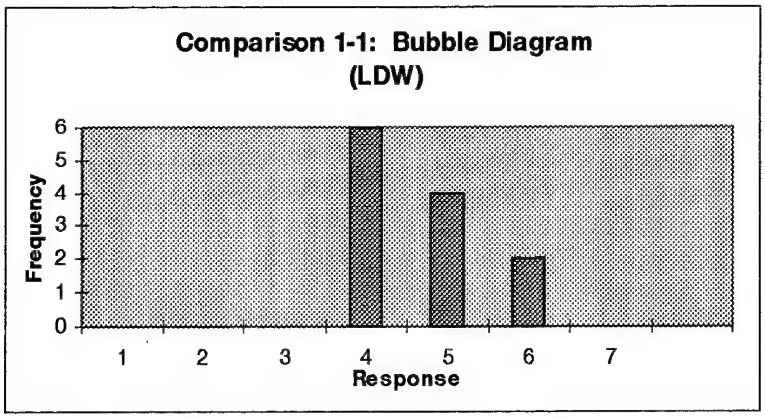


Figure 53. Comparison 1-1 LDW Responses

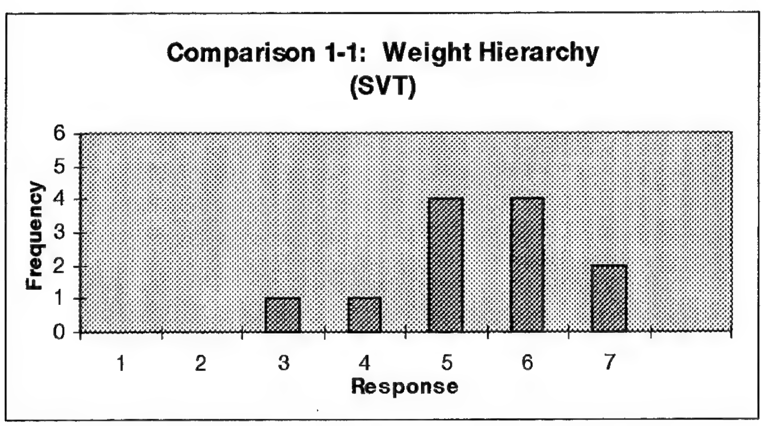


Figure 54. Comparison 1-1 SVT Responses

Comparison 1-2: Goals Hierarchy and Weight Hierarchy

Table 8. Comparison 1-2: Paired t-Test

Comparison 1-2	Goals Hierarchy	Weight Hierarchy
Mean	4.58	6.17
Variance	1.90	0.52
t Stat	-5.06	
p-value	0.00037	
t Critical two-tail	2.20	**REJECT**

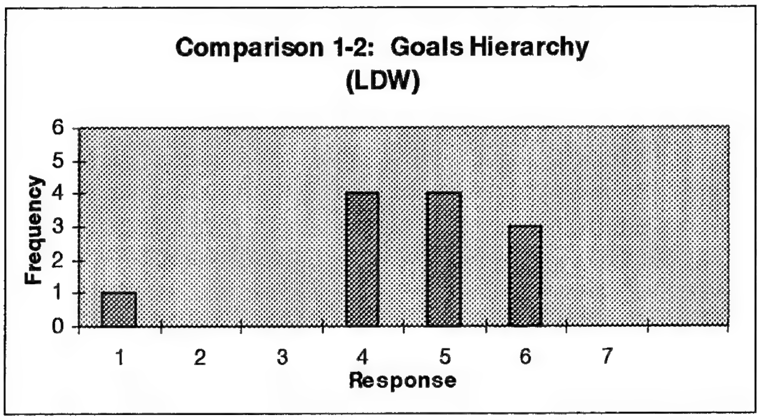


Figure 55. Comparison 1-2 LDW Responses

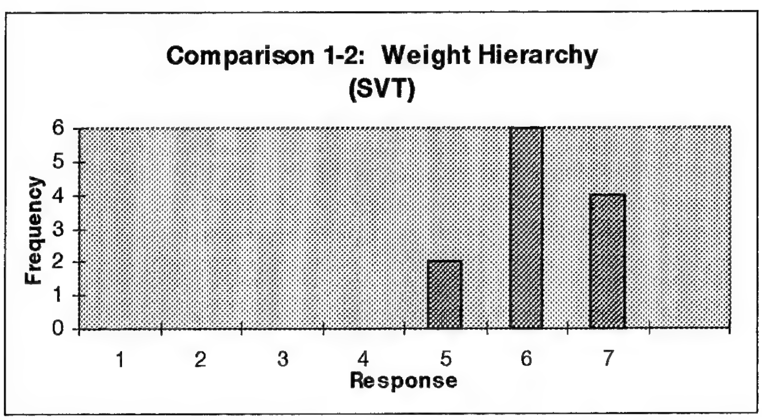


Figure 56. Comparison 1-2 SVT Responses

Comparison 2-1: Rank Alternatives and 3-D Scatter Plot

Table 9. Comparison 2-1: Paired t-Test

Comparison 2-1	Rank Alternatives	3-D Scatter Plot
Mean	4.42	5.42
Variance	3.17	1.36
t Stat	-1.51	**FAIL TO REJECT**
p-value	0.16	
t Critical two-tail	2.20	

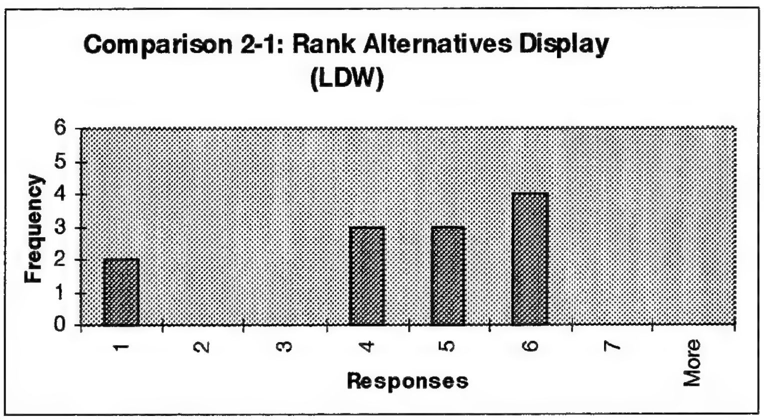


Figure 57. Comparison 2-1 LDW Responses

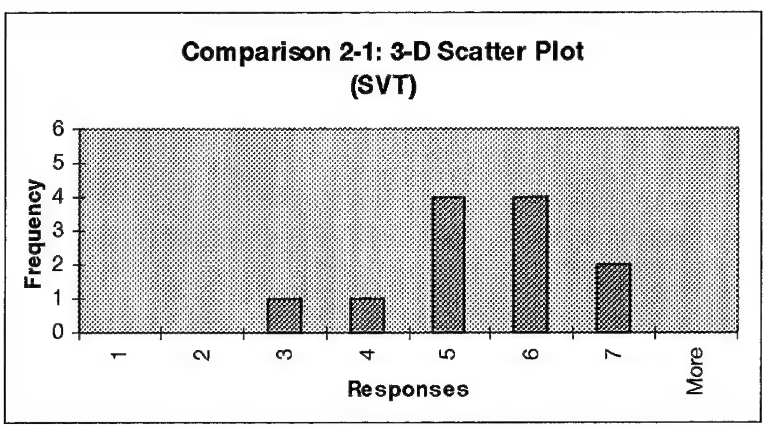


Figure 58. Comparison 2-1 SVT Responses

Comparison 2-2: Scatter Diagram and 3-D Scatter Plot

Table 10. Comparison 2-2: Paired t-Test

Comparison 2-2	Scatter Diagram	3-D Scatter Plot
Mean	4.25	5.33
Variance	2.20	1.52
t Stat	-3.03	**REJECT**
p-value	0.01	
t Critical two-tail	2.20	

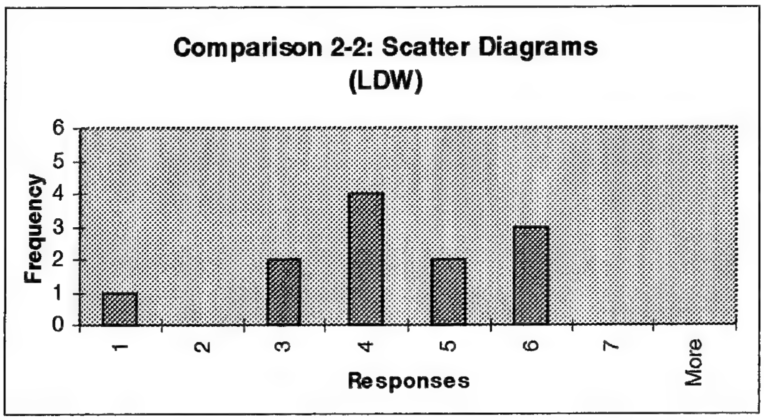


Figure 59. Comparison 2-2 LDW Responses

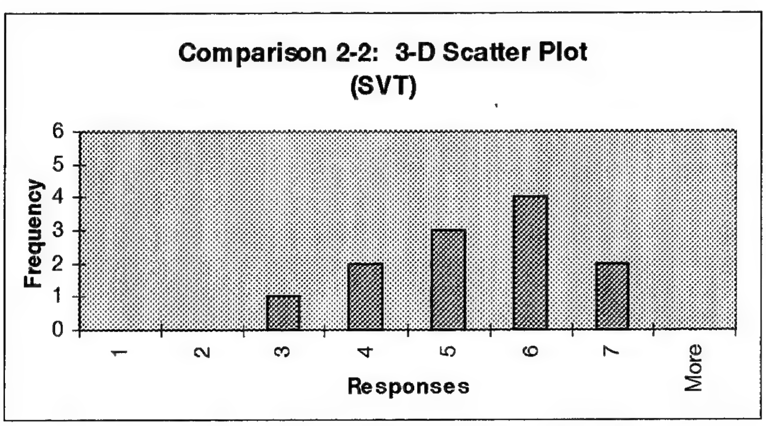


Figure 60. Comparison 2-2 SVT Responses

Comparison 3-1: Rank Alternatives and Adv 3-D Scatter Plot

Table 11. Comparison 3-1: Paired t-Test

Comparison 3-1	Rank Alternatives	Adv 3-D Scatter Plot
Mean	4.58	5.25
Variance	2.63	1.30
t Stat	-1.30	FAIL TO REJECT
p-value	0.22	
t Critical two-tail	2.20	

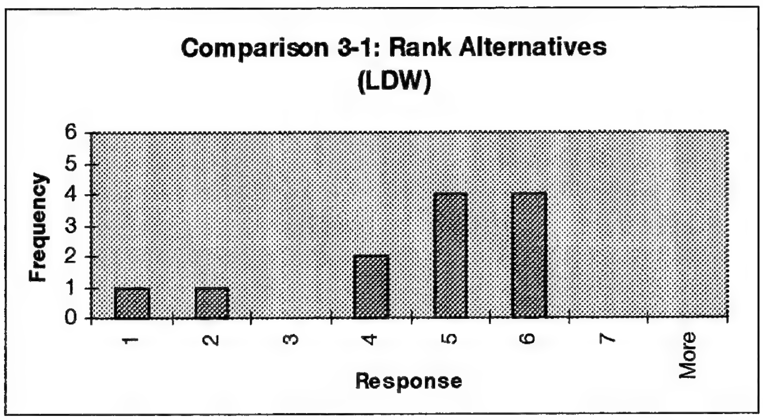


Figure 61. Comparison 3-1 LDW Responses

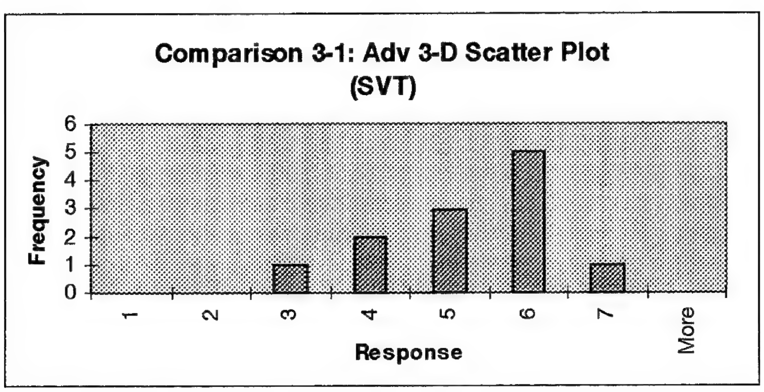


Figure 62. Comparison 3-1 SVT Responses

Comparison 3-2: Scatter Diagram and Adv 3-D Scatter Plot

Table 12. Comparison 3-2: Paired t-Test

Comparison 3-2	Scatter Diagram	Adv 3-D Scatter Plot
Mean	4.92	5.50
Variance	2.08	0.82
t Stat	-1.74	
p-value	0.11	**FAIL TO REJECT**
t Critical two-tail	2.20	

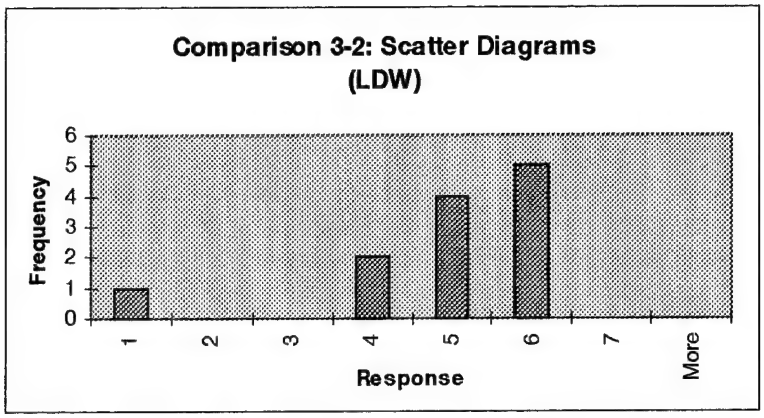


Figure 63. Comparison 3-2 LDW Responses

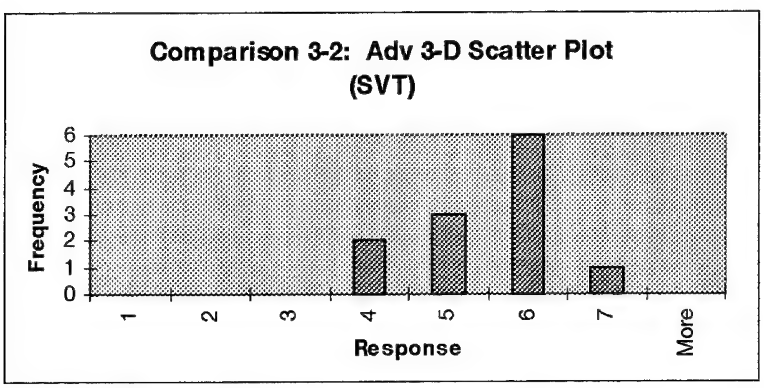


Figure 64. Comparison 3-2 SVT Responses

Comparison 4-1: Rank Alternatives and Goal Display

Table 13. Comparison 4-1: Paired t-Test

Comparison 4-1	Rank Alternatives	Goal Display
Mean	4.83	5.08
Variance	1.79	0.99
t Stat	-0.51	FAIL TO REJECT
p-value	0.62	
t Critical two-tail	2.20	

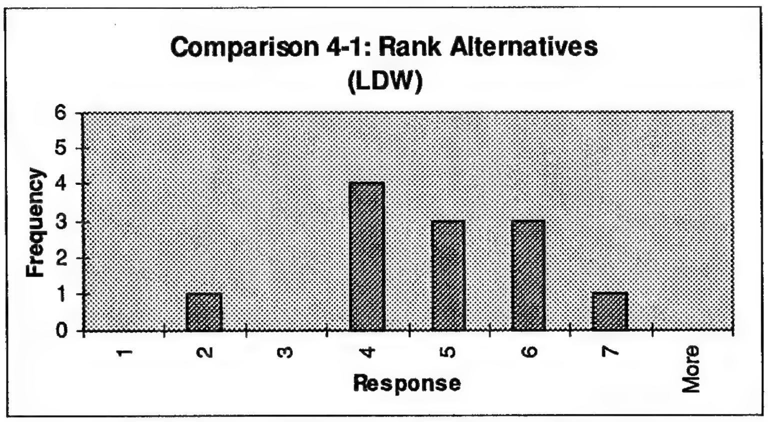


Figure 65. Comparison 4-1 LDW Responses

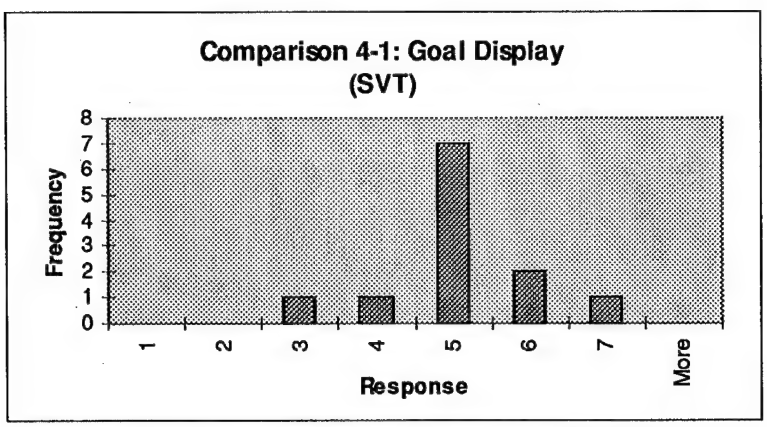


Figure 66. Comparison 4-1 SVT Responses

Comparison 4-2: Stack Bar Ranking and Goal Display

Table 14. Comparison 4-2: Paired t-Test

Comparison 4-2	Stack Bar Ranking	Goal Display
Mean	4.92	5.25
Variance	1.54	1.11
t Stat	-0.84	FAIL TO REJECT
p-value	0.42	
t Critical two-tail	2.20	

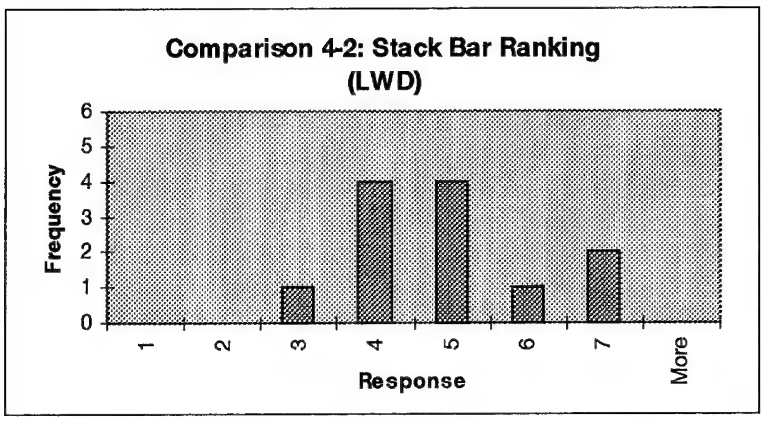


Figure 67. Comparison 4-2 LDW Responses

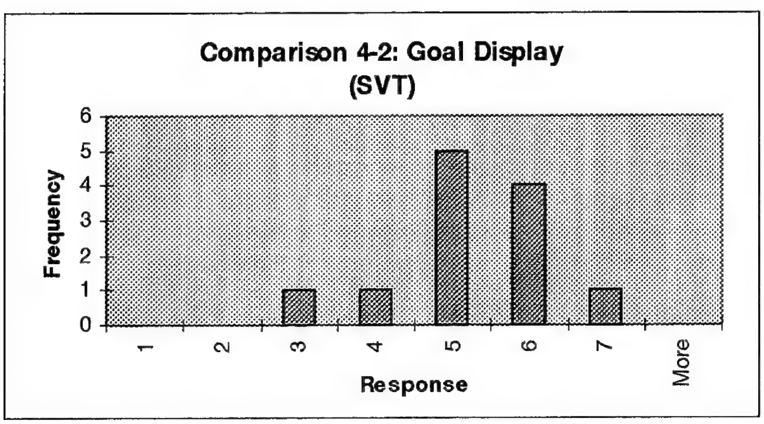


Figure 68. Comparison 4-2 SVT Responses

Comparison 4-3: Graph an Alternative and Goal Display

Table 15. Comparison 4-3: Paired t-Test

Comparison 4-3	Graph an Alternative	Goal Display
Mean	4.67	5.08
Variance	1.52	1.17
t Stat	-1.24	FAIL TO REJECT
p-value	0.24	
t Critical two-tail	2.20	

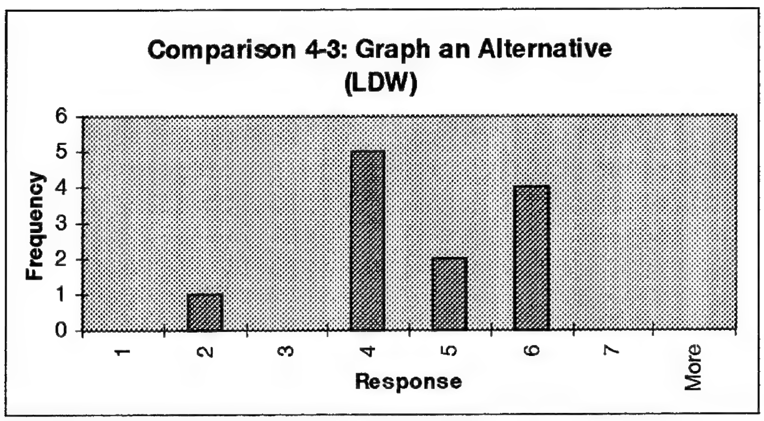


Figure 69. Comparison 4-3 LDW Responses

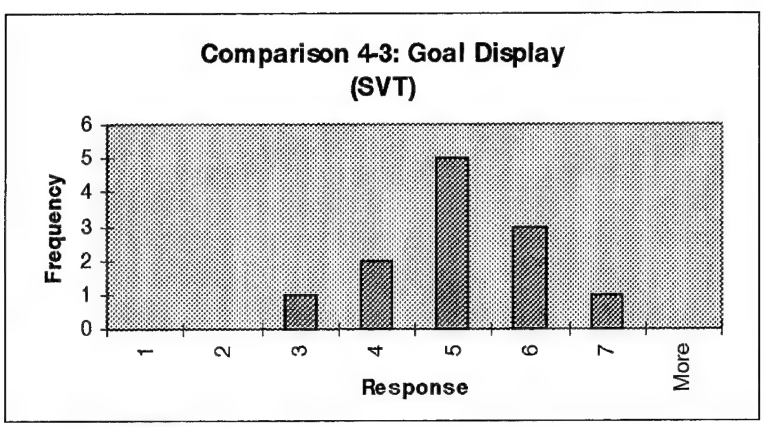


Figure 70. Comparison 4-3 SVT Responses

Comparison 4-4: Compare Alts Graph and Goal Display

Table 16. Comparison 4-4: Paired t-Test

Comparison 4-4	Compare Alts Graph	Goal Display
Mean	5.67	4.83
Variance	0.79	1.79
t Stat	2.16	**FAIL TO REJECT**
p-value	0.054	
t Critical two-tail	2.20	

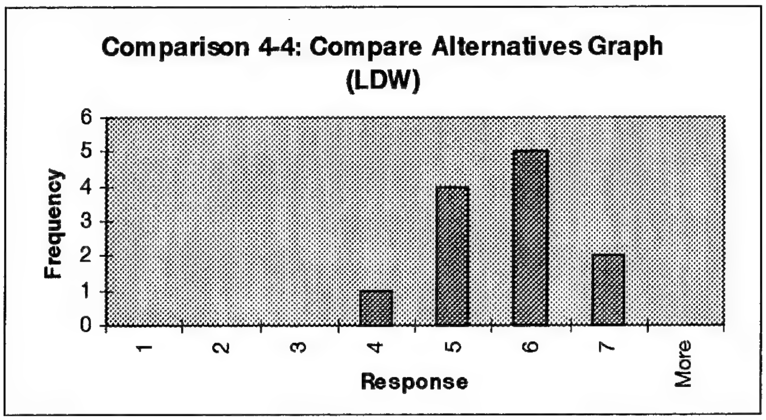


Figure 71. Comparison 4-4 LDW Responses

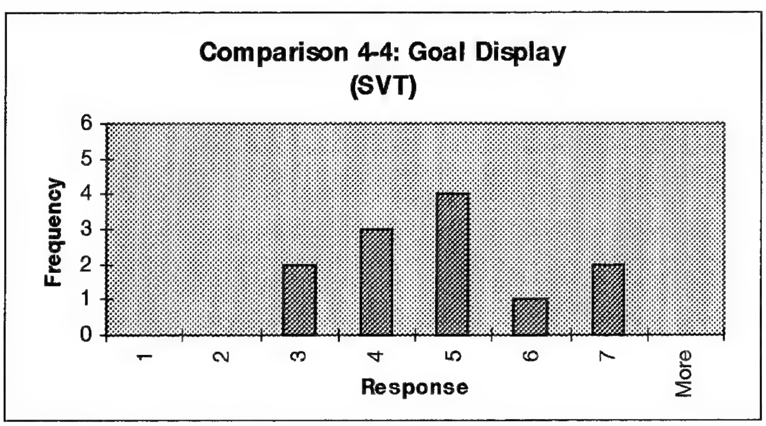


Figure 72. Comparison 4-4 SVT Responses

Comparison 5-1: Dynamic Sensitivity and Sensitivity Analysis Goal

Table 17. Comparison 5-1: Paired t-Test

Comparison 5-1	Dynamic Sensitivity	Sensitivity Analysis Goal
Mean	4.42	5.58
Variance	1.36	0.99
t Stat	-3.39	REJECT
p-value	0.01	
t Critical two-tail	2.20	

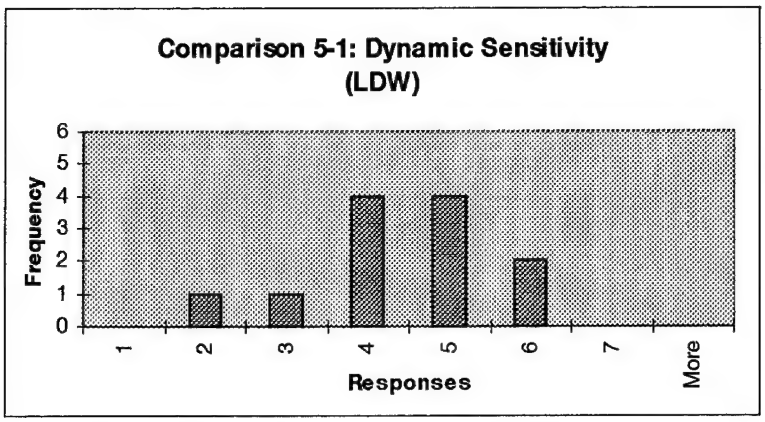


Figure 73. Comparison 5-1 LDW Responses

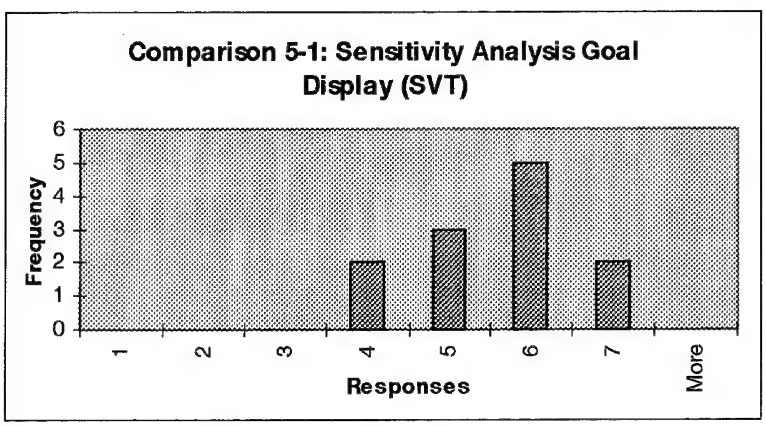


Figure 74. Comparison 5-1 SVT Responses

Comparison 5-2: Sensitivity Graph and Sensitivity Analysis Goal

Table 18. Comparison 5-2: Paired t-Test

Comparison 5-2	Sensitivity Graph	Sensitivity Analysis Goal
Mean	5.17	5.67
Variance	1.06	1.15
t Stat	-1.39	
p-value	0.19	**FAIL TO REJECT**
t Critical two-tail	2.20	

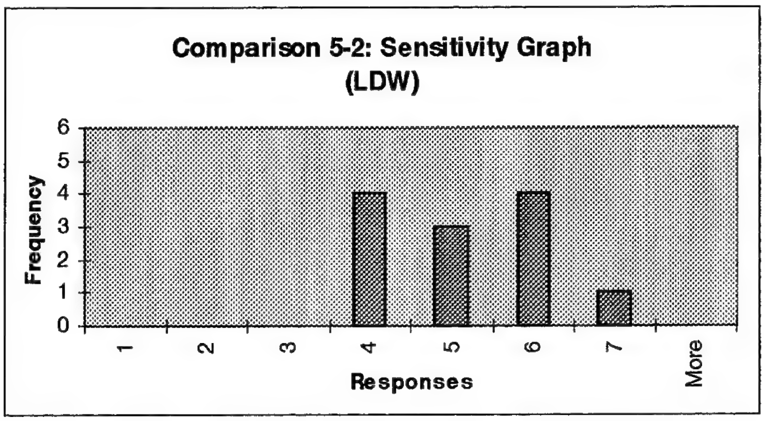


Figure 75. Comparison 5-2 LDW Responses

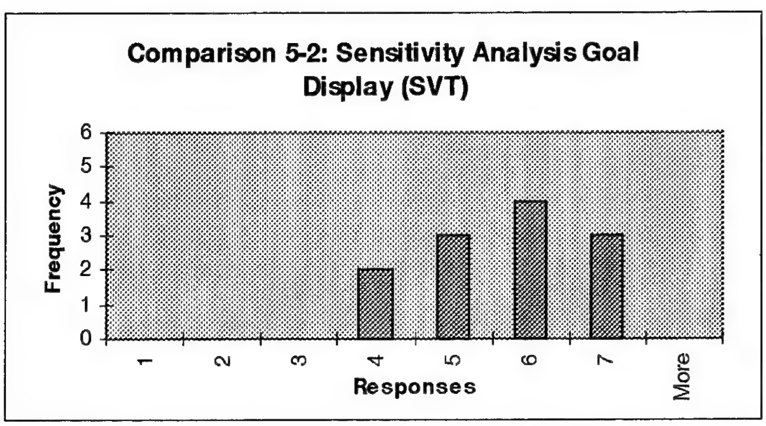


Figure 76. Comparison 5-2 SVT Responses

Comparison 5-3: Sensitivity Table and Sensitivity Analysis Goal

Table 19. Comparison 5-3: Paired t-Test

Comparison 5-3	Sensitivity Table	Sensitivity Analysis Goal
Mean	4.33	5.67
Variance	1.33	1.15
t Stat	-4.30	**REJECT**
p-value	0.0012	
t Critical two-tail	2.20	

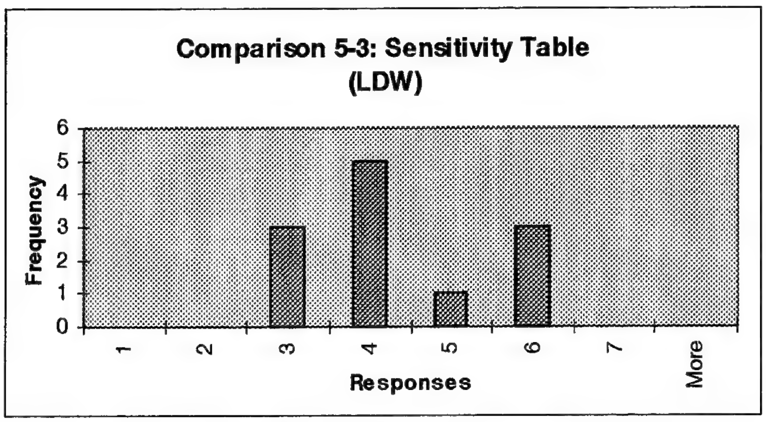


Figure 77. Comparison 5-3 LDW Responses

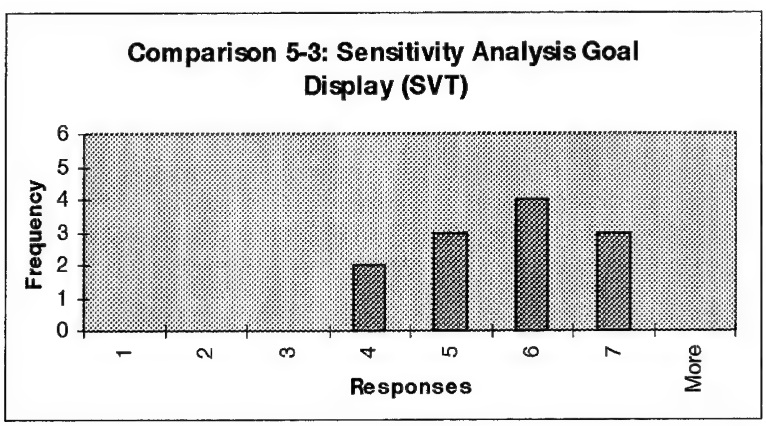


Figure 78. Comparison 5-3 SVT Responses

Comparison 6-1: Rank Alternatives and Measure Display

Table 20. Comparison 6-1: Paired t-Test

Comparison 6-1	Rank Alternatives	Measure Display
Mean	5.00	5.00
Variance	2.55	1.27
t Stat	0.00	**FAIL TO REJECT**
p-value	1.00	
t Critical two-tail	2.20	

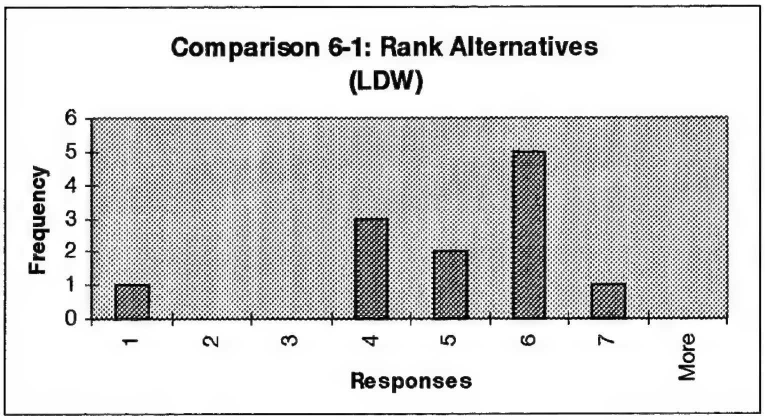


Figure 79. Comparison 6-1 LDW Responses

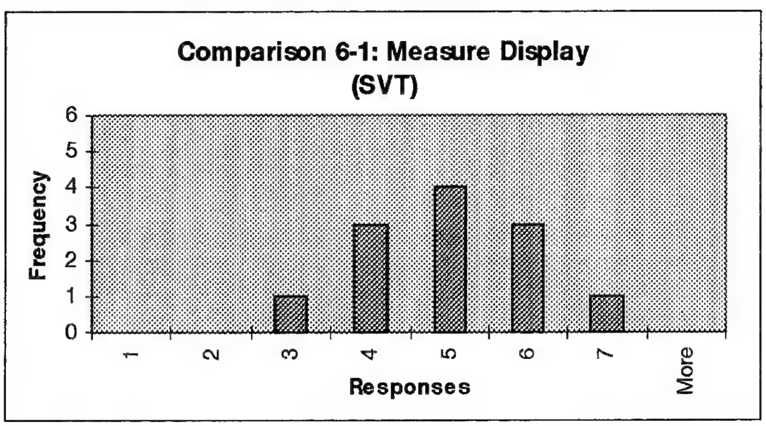


Figure 80. Comparison 6-1 SVT Responses

Comparison 6-2: Stack Bar Ranking and Measure Display

Table 21. Comparison 6-2: Paired t-Test

Comparison 6-2	Stack Bar Ranking	Measure Display
Mean	4.58	5.17
Variance	2.27	1.24
t Stat	-1.40	FAIL TO REJECT
p-value	0.19	
t Critical two-tail	2.20	

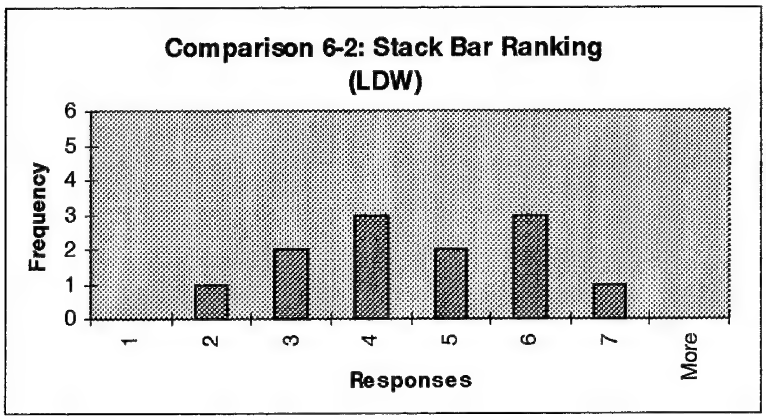


Figure 81. Comparison 6-2 LDW Responses

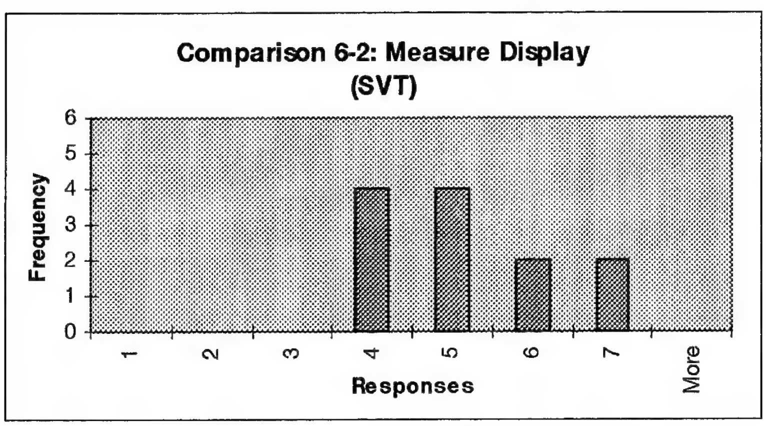


Figure 82. Comparison 6-2 SVT Responses

Comparison 6-3: Graph an Alternative and Measure Display

Table 22. Comparison 6-3: Paired t-Test

Comparison 6-3	Graph an Alternative	Measure Display
Mean	5.75	5.25
Variance	0.93	1.11
t Stat	1.91	FAIL TO REJECT
p-value	0.08	
t Critical two-tail	2.20	

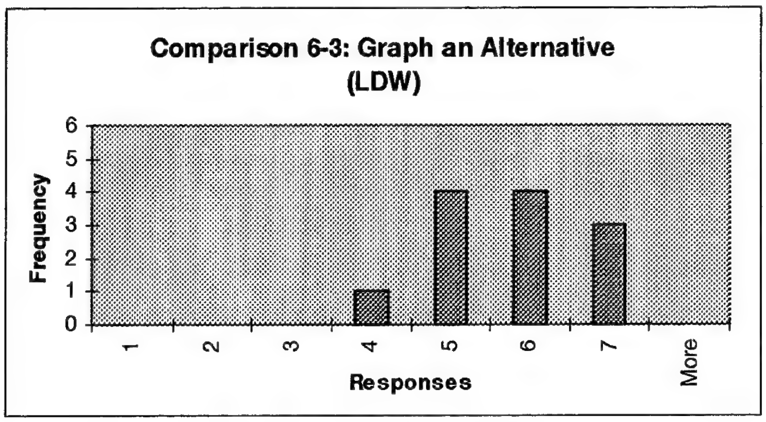


Figure 83. Comparison 6-3 LDW Responses

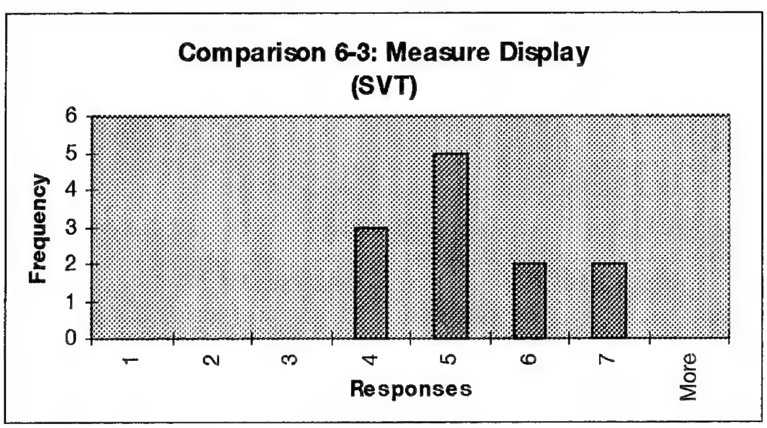


Figure 84. Comparison 6-3 SVT Responses

Comparison 6-4: Compare Alts Graph and Measure Display

Table 23. Comparison 6-4: Paired t-Test

Comparison 6-4	Compare Alts Graph	Measure Display
Mean	5.33	5.17
Variance	1.70	1.24
t Stat	0.46	**FAIL TO REJECT**
p-value	0.66	
t Critical two-tail	2.20	

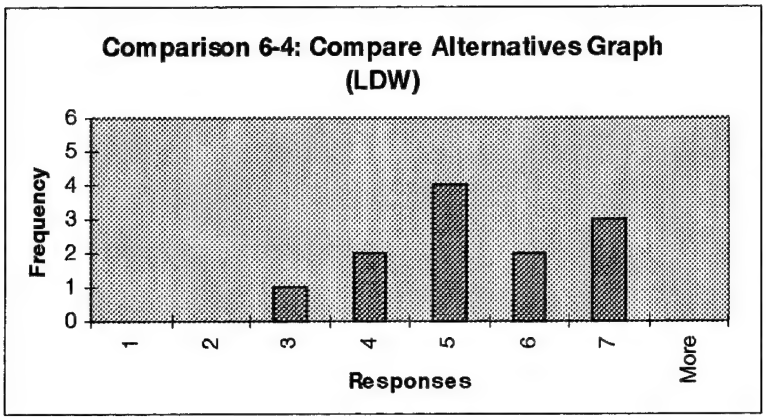


Figure 85. Comparison 6-4 LDW Responses

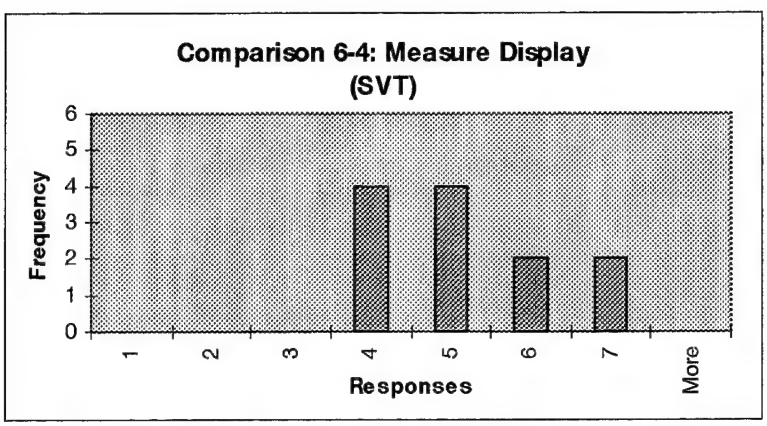


Figure 86. Comparison 6-4 SVT Responses

Comparison 7-1: Graph an Alternate and Animated Alternative

Table 24. Comparison 7-1: Paired t-Test

Comparison 7-1	Graph an Alternative	Animated Alternative
Mean	5.08	5.17
Variance	1.54	1.79
t Stat	-0.18	FAIL TO REJECT
p-value	0.86	
t Critical two-tail	2.20	

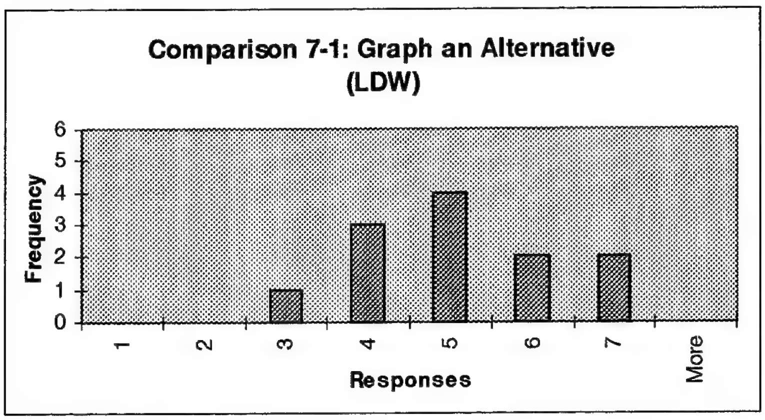


Figure 87. Comparison 7-1 LDW Responses

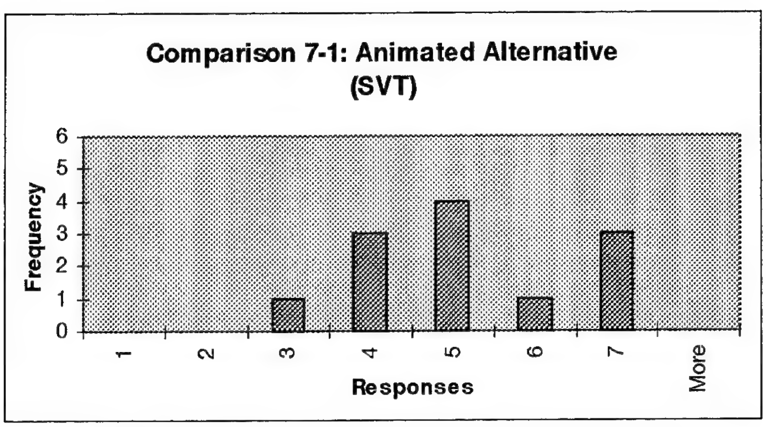


Figure 88. Comparison 7-1 SVT Responses

Comparison 7-2: Compare Alts Graph and Animated Alternative

Table 25. Comparison 7-2: Paired t-Test

Comparison 7-2	Compare Alts Graph	Animated Alternative
Mean	5.50	5.08
Variance	1.36	1.90
t Stat	0.79	**FAIL TO REJECT**
p-value	0.45	
t Critical two-tail	2.20	

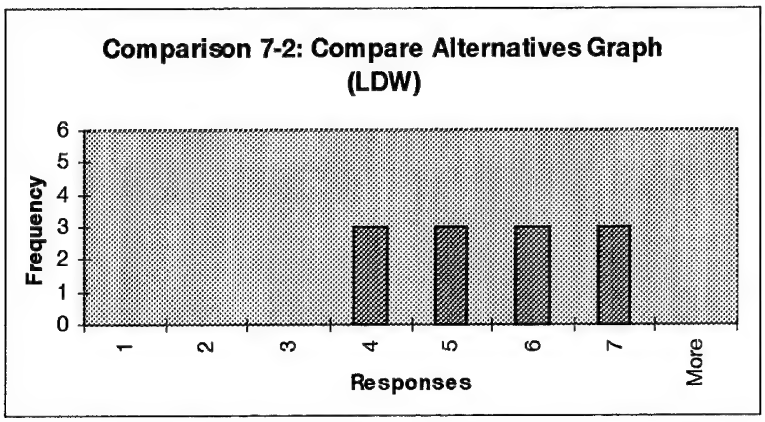


Figure 89. Comparison 7-2 LDW Responses

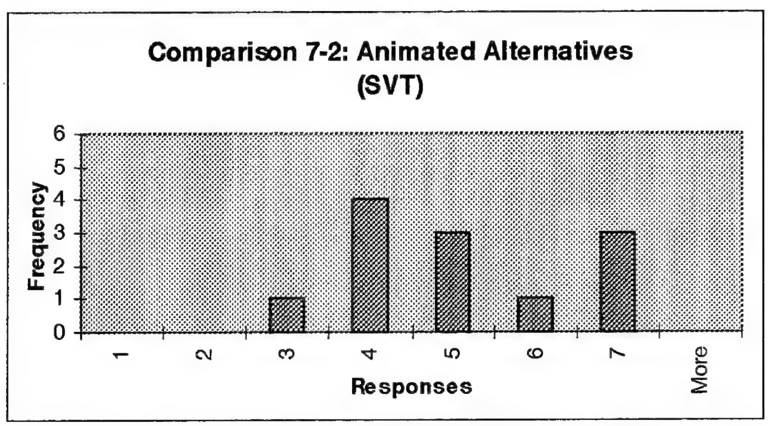


Figure 90. Comparison 7-2 SVT Responses

Comparison 8-1: Scatter Diagram and Animated Measure

Table 26. Comparison 8-1: Paired t-Test

Comparison 8-1	Scatter Diagram	Animated Measure
Mean	5.17	5.17
Variance	1.24	1.24
t Stat	0.00	
p-value	1.00	**FAIL TO REJECT**
t Critical two-tail	2.20	

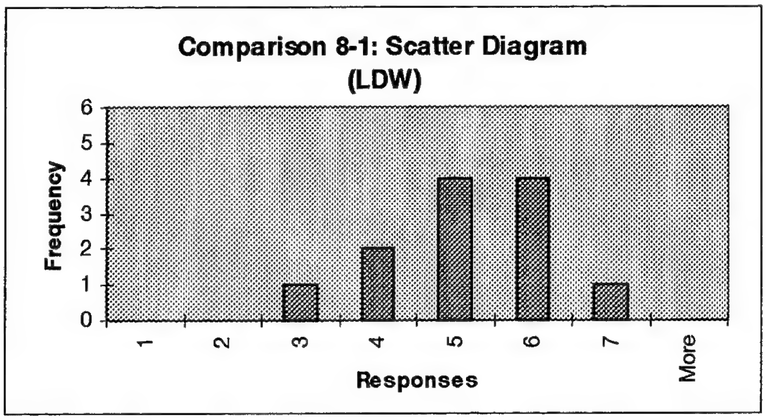


Figure 91. Comparison 8-1 LDW Responses

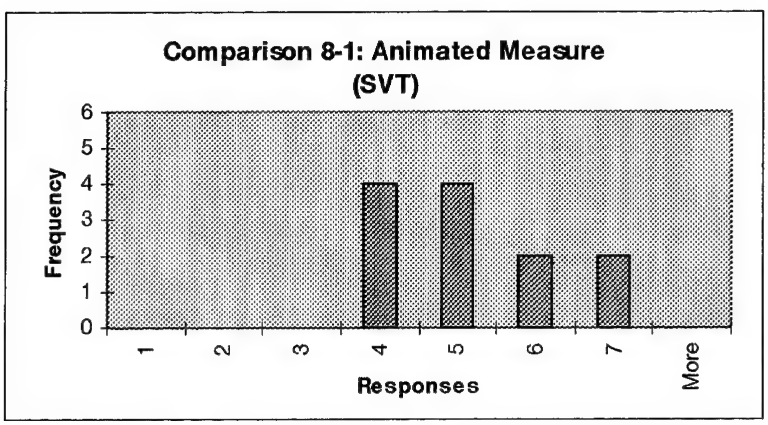


Figure 92. Comparison 8-1 SVT Responses

Comparison Overall: LDW and SVT

Table 27. Comparison Overall: Paired t-Test

Comparison - Overall	LDW	SVT
Mean	5.25	5.58
Variance	1.30	1.17
t Stat	-0.80	FAIL TO REJECT
p-value	0.44	
t Critical two-tail	2.20	

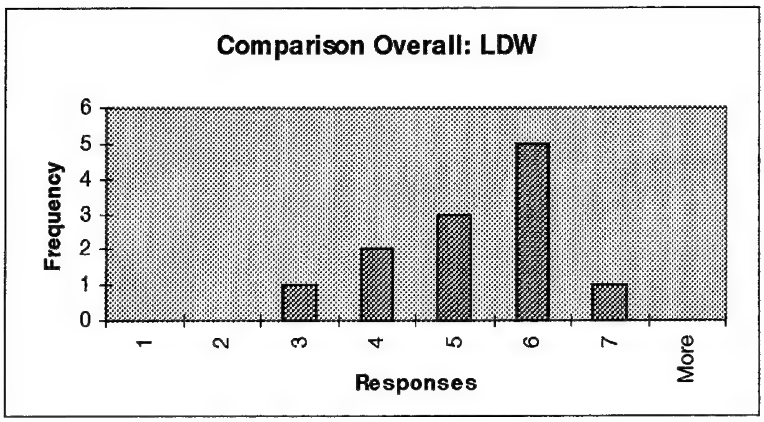


Figure 93. Comparison Overall LDW Responses

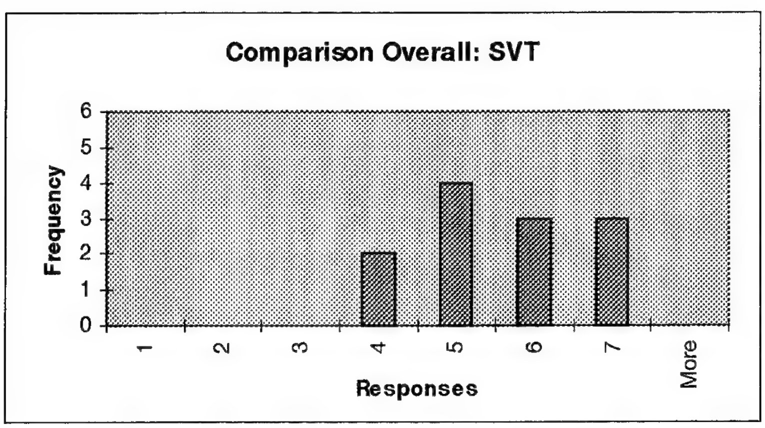


Figure 94. Comparison Overall SVT Responses

Appendix C: Survey

Purpose: The purpose of this survey is to validate the effectiveness of a software visualization tool created to enhance an operations research analysis.

Introduction: The process consists of comparing several interactive three-dimensional visualizations created by the software visualization tool with similar displays generated with a currently used operations research analysis software package. The survey will initially ask you for some information regarding your knowledge of decision analysis and value focused thinking, and whether you are familiar with the software being demonstrated. Then, for each comparison the survey will ask for a quantitative response based upon your perceived effectiveness of both displays. The range for your response is scaled from 1 to 7 using the following guidelines:

1: Visualization was not effective; provided basically no insight into the analysis
4: Visualization was moderately effective; provided some insight into the analysis
7: Visualization was extremely effective; provided significant insight into the analysis

You will then circle the appropriate response. In addition, for each comparison the survey will also ask "what other purpose the visualization might be useful for?" and "how would you improve this visualization?".

Data Description: The data used in this analysis is multivariate and hierarchical. The value focused model consisted of 23 alternatives (or trains) for a decision maker to select from. For each train, there is a corresponding objective value. This value represents a goal and is at the highest level within the hierarchy. Its value is a weighted linear combination of the subgoals beneath it. The hierarchy continues downward until a goal consist of a linear combination of measures, which are at the lowest level of the hierarchy. Each goal and measure has an associated weight and utility value.

General Questions:

Have you taken any courses in decision analysis and/or value-focused thinking? _____

Are you familiar with or have you ever used Logical Decisions for Windows? _____

Have you taken any courses in multivariate data analysis? _____

Comparison 1: How effective are the displays in representing the overall structure of the data as well as identifying the relative contribution (importance) of each subgoal and measure?

LDW(9):	1	2	3	4	5	6	7	SVT(1):	1	2	3	4	5	6	7
LDW(10):	1	2	3	4	5	6	7	SVT(1):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 2: How effective are the displays in representing the influence of each of the 5 criteria goals with respect to the overall utility?

LDW(1):	1	2	3	4	5	6	7	SVT(2):	1	2	3	4	5	6	7
LDW(6):	1	2	3	4	5	6	7	SVT(2):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 3: How effective are the displays in representing the influence of each of the 5 criteria goals with respect to the overall utility?

LDW(1):	1	2	3	4	5	6	7	SVT(3):	1	2	3	4	5	6	7
LDW(6):	1	2	3	4	5	6	7	SVT(3):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 4: How effective are the displays in comparing the performance of the alternatives for selected goals?

LDW(1):	1	2	3	4	5	6	7	SVT(4):	1	2	3	4	5	6	7
LDW(2):	1	2	3	4	5	6	7	SVT(4):	1	2	3	4	5	6	7
LDW(7):	1	2	3	4	5	6	7	SVT(4):	1	2	3	4	5	6	7
LDW(8):	1	2	3	4	5	6	7	SVT(4):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 5: With respect to performing sensitivity analysis (changing the weight) on the five criteria goals, how effective are the visual displays in representing this new information?

LDW(3):	1	2	3	4	5	6	7	SVT(5):	1	2	3	4	5	6	7
LDW(4):	1	2	3	4	5	6	7	SVT(5):	1	2	3	4	5	6	7
LDW(5):	1	2	3	4	5	6	7	SVT(5):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 6: How effective are the displays in comparing the performance of the alternatives for selected measures?

LDW(1):	1	2	3	4	5	6	7	SVT(6):	1	2	3	4	5	6	7
LDW(2):	1	2	3	4	5	6	7	SVT(6):	1	2	3	4	5	6	7
LDW(7):	1	2	3	4	5	6	7	SVT(6):	1	2	3	4	5	6	7
LDW(8):	1	2	3	4	5	6	7	SVT(6):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 7: How effective are the displays in comparing two alternatives, based upon their respective measure scores?

LDW(7):	1	2	3	4	5	6	7	SVT(7):	1	2	3	4	5	6	7
LDW(8):	1	2	3	4	5	6	7	SVT(7):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

Comparison 8: How effective are the displays in comparing two measures, based upon the values of all 23 alternatives?

LDW(6):	1	2	3	4	5	6	7	SVT(8):	1	2	3	4	5	6	7

What other purpose might the visualization be useful for?

How would you improve this visualization?

OVERALL: I felt the amount of user interaction provided with the software displays (being able to control certain features such as viewpoint, number of variables displayed, color, etc) helped increase my overall understanding of the data set and any analysis performed.

LDW:	1	2	3	4	5	6	7	SVT:	1	2	3	4	5	6	7

FINAL COMMENTS:

Survey Information - LDW Displays

SVT	1	2	3	4	5	6	7	8
LDW	Default	3DSP	A3DSP	GOAL	SAG	MEASURE	AA	AM
1 - Rank Alternative		X	X	X		X		
2 - Stack Bar Ranking				X		X		
3 - Dynamic Sensitivity					X			
4 - Sensitivity Graph					X			
5 - Sensistivity Table					X			
6 - Scatter Diagram		X	X					X
7 - Graph an Alternative				X		X	X	
8 - Compare Alternatives Graph				X		X	X	
9 - Bubble Diagram	X							
10 - Goals Hierarchy	X							

Rank Alternatives: shows a ranking of the alternatives based on any of the goals or measures

Stacked Bar Ranking: shows the overall ranking results for the analysis

Dynamic Sensitivity: can see the effects of changes in the weights for the goals and measures

Sensitivity Graphs: displays the effect of varying a measure or goal's weight from 0 to 100 percent

Sensitivity Table: review an overall ranking based on any percentage of weight on a goal or measure

Scatter Diagrams: compare the performance of the alternatives on any two measures or goals. One measure or goal on each axis.

Graph an Alternative: displays a bar graph showing the performance of a single alternative on the measures or goals. Its width of the bar for a measure or goal is proportional to its weight.

Alternative Uncertainty Graph: view a graph of the uncertainty in a single alternative for a single measure or goal.

Compare Alternatives Graph: shows a detailed comparison of the differences between two alternatives.

Appendix D: Software Class Listing

Section I: Spacecrafter Classes

Name:
CheckerTerrain

Constructor:
```
CheckerTerrain(
short nx,short nz,
float lengthx,float lengthz,
int color1,int color2,
boolean bothsides
)
```

Input Parameters:

nx	number of cells on x axis
nz	number of cells on z axis
lengthx	cell length on x axis
lengthz	cell length on z axis
color1	hexidecimal color constant for odd cells
color2	hexidecimal color constant for even cells
bothsides	true if object is double sided, otherwise false

Return Value:
A valid CheckerTerrain object on success, null on failure

Class Variables:
None

Class Methods:
None

****

Name:
Cube

Constructor:
```
Cube(float width,float height,float depth,boolean bothsides)
```

Input Parameters:

width	width of cube(x-axis)
height	height of cube(y-axis)
depth	depth of cube(z-axis)
bothsides	true if object is double sided, otherwise false

Return Value:

A valid Cube Object on success, null on failure

Class Variables:

subclass of Object3D

Class Methods:

subclass of Object3D

****

Name:

Light3D (FINAL CLASS)

Constructor:

Light3D(byte Type)

Input Parameters:

Type

POINTLIGHT	Point light
CONELIGHT	Conical light(NOT SUPPORTED AT THIS TIME)
SPOTLIGHT	Spot light(NOT SUPPORTED AT THIS TIME)

Return Value:

None

Class Variables:

Point3D Pos	Light's current world position
Point3D Dir	Light's current directional vector
boolean IsOn	Is the light on?
float Intensity	Light's current intensity setting

Methods:

getDirection
getPosition
setDirection
setDirectionDirect
setIntensity
setOnOff
setPosition
setPositionDirect

****

Name:

Matrix4X4 (FINAL CLASS)

Constructor:

Matarix4X4()

Input Parameters:

None

Return Value:

A valid Matrix4X4 object on success, null on failure

Class Variables:

float M4x4[4][4]   4 x 4 array of floats representing orientation

Class Methods:

copy4X4
eulerToMatrix
identity4X4
inverseFast4X4
mult4X4
zero4X4

****

Name:

Object3D

Constructor:

Object3D()

Note that since the Object3D class is extended by the actual 3D object type (cube, cone, sphere, etc.), it is not necessary to call the Object3D constructor (unless specifically documented by a 3'rd party implementation) as it is called automatically by the extended class type's constructor. Also, World3D object must be created before creating any Object3D objects. Failure to do so will result in unpredictable results.

Input Parameters:

None

Return Value:

A valid 3D Object on success, otherwise null on failure

Class Variables:

short ID	Internal object ID - Reserved but Available
short Type	Object type - Cube, User defined, etc?
short Num_Vertices	Number of vertices in object
short Num_Polygons	Number of polygons in object
Point3D[] Vertices_Local	Array of local vertices for object
Polygon3D[] Polygon	Array of polygons for object
Point3D WorldPos	World position of object
Matrix4X4 Orientation	Local orientation of object
Point3D PivotPoint	Pivot point of object
Point3D Extent	Extents of object
boolean InSim	Is object in the current simulation list?
boolean Visible	Is object currently visible?
byte DrawFlags	Reserved
Object3D Parent	Object's parent
Vector Child	Java Vector containing object's children references

Class Methods:

addToSimulation
attach
collide
detach
draw
finishBuild
getAxis
getChild
getExtents
getNextObject
getNumPolys
getOrientationMatrix
getPixelSize
getPoly
getPosition
getVisibility
inSimulation
prepForBuild
removeChildren
removeFromSimulation
resetTexture
resetTextureFlags
rotate
rotateAroundPoint
rotateMultiple
scale
scaleMultiple

setCentroid
setColor
setCullMode
setDoubleSided
setDrawMode
setOrientationMatrix
setPivot
setPosition
setPositionDirect
setTexture
setVisibility
shiftLocalsToCenter
translate

****

Name:
Picker2D3D

The Picker2D3D class lets you pick 3D objects in a simulation using 2D screen coordinates. This is very usefull for 'grabbing' objects and end-user manipulation of objects.

Constructor:
Picker2D3D()

Input Parameters:
None

Return Value:
A valid Picker2D3D object if successful, otherwise null

Class Variables:
None

Class Methods:
None

****

Name:
Point3D (FINAL CLASS)

Constructor:
Point3D()
Point3D(float Val)

Input Parameters:

Val - If using second constuctor method the value all elements are set to initially

Return Value:

A Valid Point3D object on success, null on failure

Class Variables:

float x,y,z,w	x,y,z position or direction - Note w is not currently used

Class Methods:

add
copy
crossProduct
distance
dotProduct
init magnitude
magnitudeSQ
makeVector
normalize
set
subtract
vectorMag

****

Name:

Sphere

Constructor:

Sphere (float radius, int lontess, int lattess, boolean bothsides)

Input Parameters:

radius	Radius of sphere(x,y,z axis)
lonetess	Longitude tesselations(edges)
lattess	Latitude tesselations(edges)
bothsides	true if object is double sided, otherwise false

Return Value:

A valid Sphere object on success, null on failure

Class Variables:

subclass of Object3D

Class Methods:

subclass of Object3D

****

Name:
Viewpoint3D (FINAL CLASS)

Constructor:
Viewpoint3D()

Input Parameters:
None

Return Value:
A Valid Viewpoint3D Object on success, otherwise null on failure

Class Variables:

Point3D Pos	Viewpoint's current world position
Matrix4X4 Orientation	Viewpoint's current local orientation

Class Methods:
getAxis
getDirection
getOrientationMatrix
getPosition
point3D_2Camera
rotate
setOrientationMatrix
setPosition
setPositionDirect
translate

****

Name:
World3D

Constructor:
World3D(Component Comp)

Input Parameters:
Comp - Component instance

Return Value:
A valid World3D object on success, null on failure

Class Variables:

long GlobalNumObjects	Total objects created in this simulation, including objects that have been removed, but not deleted
long GlobalNumPolys	Total polygons created in this simulation, including polygons to objects that have been removed, but not deleted
long GlobalNumVerts	Total vertices created in this simulation, including vertices to objects and polygons that have been removed, but not deleted
Point3D GlobalWorldExtents	Complete world extents, expressed as a positive value
Point3D GlobalWorldExtentsLo	Low value extents of the world
Point3D GlobalWorldExtentsHi	High value extents of the world
int Clipping3DType	The current clipping type setting
float AmbientLight	The current ambient light intensity setting

Class Methods:

buildRampEntry
buildRampIndex
displayIt
draw3DObjects
flushPolyList
flushPolyListNoReset
getBoundaries
getBufferPtr
getExtents
getImagePtr
getStats
modifyRampEntry
modifyRampIndex
num3DObjectsInList
prepForDisplay
prepForRendering
prepForUnwrap
realizeNewPalette
resetPolyList
setAmbientLight
setBackgroundColor
setDrawMode
setHitherYon
setPaletteEntriesPerColor
setViewpoint
setWorld
setZSortMode

set3DClipType
set3DWindowDimensions

World3D CoreSpace Interface Methods:
getFlatShadedTriangleHandle
getGraphicsDeviceHandle
getGraphics2DDeviceHandle
getPolygonSortingHandle
getTexturedTriangleHandle
setFlatShadedTriangleEngine
setGraphicsDeviceEngine
setGraphics2DDeviceEngine
setPolygonSortingEngine
setTexturedTriangleEngine

Section II: Thesis created Classes/Subclasses

Name:
Goal

Constructor:
Goal()

Input Parameters:
None

Return Value:
A goal object

Class Variables:

abs_utility_value	the product of the aggregated weight and the utility value
abs_weight_value	the product of all aggregated weights
utility_value	the actual value associated with the goal
weight_value	the weight assigned
wt_utility_value	the product of the value and the weight

Class Methods:
modifyWeight
setUtilityValue
setWeight
updateAbsUtilityValue
updateAbsWeightValue
updateWtUtilityValue

****

Name:
Measure

Constructor:
Measure()

Input Parameters:
None

Return Value:
A measure object

Class Variables:
raw_utility   the raw value score
subclass of Goal

Class Methods:

computeUtilityValue
setRawUtility
subclass of Goal

****

Name:

MyCheckerTerrain

Constructor:

MyCheckerTerrain(
int num_x_cells, int num_z_cells,
float x_length, float z_length,
Color c_odd, Color c_even)

Input Parameters:

num_x_cells	number of cells on x axis
nun_z_cells	number of cells on z axis
x_length	cell length on x axis
z_length	cell length on z axis
color1	color object for odd cells
color2	color object for even cells

Return Value:

A valid CheckerTerrain object on success, null on failure

Class Variables:

subclass of CheckerTerrain

Class Methods:

subclass of CheckerTerrain

****

Name:

MyCube

Constructor:

MyCube(float x, float y, float z)

Input Parameters:

x	width of cube(x-axis)
y	height of cube(y-axis)
z	depth of cube(z-axis)

Return Value:
A valid Cube Object on success, null on failure

Class Variables:

x_size	width of cube(x-axis)
y_size	height of cube(y-axis)
z_size	depth of cube(z-axis)
rescale_x	inverse of the x scale multiple
rescale_y	inverse of the y scale multiple
rescale_z	inverse of the z scale multiple
color_value	color object to represent the color of the cube
subclass of Cube	

Class Methods:
setColor
getColor
setScaleFactors
getXRescale
getYRescale
getZRescale
subclass of Cube

****

Name:
MyPicker2D3D

Constructor:
MyPicker2D3D()

Input Parameters:
None

Return Value:
A valid Picker2D3D object if successful, otherwise null

Class Variables:
subclass of Picker2D3D

Class Methods:
subclass of Picker2D3D

****

Name:

MySphere

Constructor:

MySphere(float x, int y, int z)

Input Parameters:

x Radius of sphere(x,y,z axis)
y Longitute tesselations(edges)
z Lattitude tesselations(edges)

Return Value:

A valid Sphere object on success, null on failure

Class Variables:

radius	radius of sphere
color_value	color of sphere

subclass of Sphere

Class Methods:

setColor
getColor
subclass of Sphere

****

Name:

MyWorld3D

Constructor:

MyWorld3D(Component Comp)

Input Parameters:

Comp - Component instance

Return Value:

A valid World3D object on success, null on failure

Class Variables:

subclass of World3D

Class Methods:

subclass of World3D

Appendix E: Software Data Dictionary

The following list identifies the included files, data structures and methods of the software visualization tool. A listing of all classes can be found in Appendix C.

IMPORTED FILES:

java.awt.*	Abstract Windowing Toolkit classes and interfaces
java.awt.Button	a user-interface push button
java.awt.Event	object representing events caused by the system or user input
java.awt.Image	abstract representation of a bitmap image
java.io.*	input and output classes
java.lang.*	classes and interfaces that are the core of Java
java.net.*	classes and interfaces for performing network operation

InWorld's Java 3D APIs and subclasses (see Appendix C):

inworld.spacecrafter.*
inworld.spacecrafter.MyCheckerTerrain
inworld.spacecrafter.MyCube
inworld.spacecrafter.MyObject3D
inworld.spacecrafter.MyPicker2D3D
inworld.spacecrafter.MySphere
inworld.spacecrafter.MyWorld3D

inworld.corespace.*

APPLET:

thesis

CONSTANTS:

int number_of_colors	number of colors - 8
int number_of_goals	number of goals - 25
int number_of_marker_spheres	number of marker spheres - 5
int number_of_measures	number of measures - 28
int number_of_sag_goals	number of SAG goals - 5
int number_of_sag_rows	number of SAG rows available for display - 5
int number_of_sp_chkboxs	number of check boxes - 7
int number_of_trains	number of trains - 23
int number_of_wt_rows	number of wt rows in external file - 2
int pts_per_train	number of points per train - 4 (AM display)
int pts_per_measure	number of points per measure - 4 (AA display)
float asp_default_size	initial size of adv scatter plot cubes - 50 units
float aac_default_size	size of animated alternative (AA) cube - 75 units
float aasc_default_size	size of AA stationary cubes - 37.5 units
float aas_default_size	size of AA sphere - 50 units
float aass_default_size	size of AA stationary spheres - 25 units
float amc_default_size	size of Animated Measure cube - 75 units
float amsc_default_size	size of AM stationary cubes - 37.5 units
float ams_default_size	size of AM sphere - 50 units
float amss_default_size	size of AM stationary sphere - 25 units
float gdc_default_size	initial size of goal display cubes - 50 units
float marker_sphere_size	size of marker spheres - 30 units(radius)
float mdc_default_size	initial size of measure display cubes - 50 units
float sag_default_size	initial size of sag display cubes - 50 units
float sp_default_size	size of scatter plot cubes - 50 units
float max_axis	maximum distance along an axis, 2000 units
float x_offset	offset for x axis , -1000 units
float x_max	maximum x axis value, 1000 units
float y_offset	offset for y axis -500 units
float y_max	maximum y axis value, 1500 units
float z_offset	offset for z axis, 0 units
float z_max	maximum z axis value, 2000 units

VARIABLES:

Color my_orange	redefine the orange color object
String file_name	file name for the data file
String help_file_name	file name for the help file
String g_string	array of goal names
String m_string	array of measure names
String t_string	array of train names
float file_data_array	array structure to read the external data into
Goal goal_object_array	2-D array to store the goals
Measure measure_object_array	2-D array to store the measures
MyCube aa_cube	animated alternative (AA) display cube
MyCube aas_cube_array	AA array of stationary cubes
MySphere aa_sphere	AA display sphere
MySphere aas_sphere_array	AA array of stationary spheres
Point3D aa_cube_point_array	AA array of display cube points
Point3D aas_cube_point_array	AA array of stationary cube points
Point3D aa_sphere_point_array	AA array of display sphere points
Point3D aas_sphere_point_array	AA array of stationary sphere points
int aa_index	index to display cube/sphere point arrays
boolean aa_display_ready	boolean flag to determine if display is ready
MyCube am_cube	animated measure (AM) display cube
MyCube ams_cube_array	AM array of stationary cubes
MySphere am_sphere	AM display sphere
MySphere ams_sphere_array	AM array of stationary spheres
Point3D am_cube_point_array	AM array of display cube points
Point3D ams_cube_point_array	AM array of stationary cube points
Point3D am_sphere_point_array	AM array of display sphere points
Point3D ams_sphere_point_array	AM array of stationary sphere points
int am_index	index to display cube/sphere point arrays
boolean am_display_ready	boolean flag to determine if display is ready
MyCube asp_cube_array	the Adv 3D scatter plot array of cubes
Point3D asp_point_array	the Adv 3D scatter plot array of points
MyCube gd_cube_array	the goal display array of cubes
MyCube hierarchy_cube_array	the hierarchy array of goal and measure cubes
MySphere marker_sphere_array	array of spheres

MyCube md_cube_array	the measure display array of cubes
MyCube sag_cube_array	the sensitivity analysis goal array of cubes
Point3D sag_cube_point_array	the SAG array of points
float new_value_array	the SAG array of computed new values
float new_weight_array	the SAG array of computed new weights
MyCube sp_cube_array	the 3D scatter plot array of cubes
Point3D sp_point_array	the 3D scatter plot array of points
int bv_index;	best value in 3D scatter plot (rotate)
int wv_index;	worst value in 3D scatter plot (rotate)
int display_selected	identifies the display selected
MyPicker2D3D object_picker	the picker object to select displayed objects
boolean object_grabbed	boolean varaible to check if an object is picked
Object3D object_picked	generic 3D object to compare against
Point3D op_pos	variable to hold object picked current's position
Matrix4X4 op_ori_matrix	matrix to hold orientation information
MyCube outline_cube	frames the selected objected
MyCheckerTerrain world_terrain	random terrain
Point3D world_terrain_pos	position of the terrain
Point3D world_terrain_hidden	terrain is hidden - AA and AM displays
int win_offset	image offset to start drawing
Point3D standard_viewpoint	contain the viewpoint
float view_speed	the navigational speed
MyWorld3D my_world	a 3D world
Viewpoint3D my_view	a 3D viewpoint
Light3D light_1, light_2	world lights
Thread runner	thread for process
boolean running	flag - is the process running?
boolean paused	pause the cubes - AA and AM displays
boolean nav_button_pressed	flag - was a navigational button pressed?
int which_nav_button_pressed	flag - which navigational button pressed?
Panel nav_button_panel	panel containing navigational buttons
Button nav_button	array of navigational buttons
Panel display_button_panel	panel containing display buttons
Button display_button	array of display buttons

CardLayout cardlayout	interface screens - series of cards in a deck
Panel cards	the interface screen panel

The following sets up the animated altenerative display interface screens. There are two panels, one to select the first trains and one to select the second train.

Panel aa1_panel	the overall panel - first train
Panel aa1_train_panel	sub panel of train choices
Panel aa1_button_panel	sub panel of buttons
Label aa1_header_label	overall header label
CheckboxGroup aa1_cbg	checkbox group - only 1 can be selected
Checkbox aa1_cb	array of train choices
Button aa1_button	array of buttons

Panel aa2_panel	the overall panel - second train
Panel aa2_train_panel	sub panel of train choices
Panel aa2_button_panel	sub panel of buttons
Label aa2_header_label	overall header label
CheckboxGroup aa2_cbg	checkbox group - only 1 can be selected
Checkbox aa2_cb	array of train choices
Button aa2_button	array of buttons

The following sets up the animated measure display interface screens. There are two panels, one to select the first measure and one to select the second measure.

Panel am1_panel	overall panel - first measure
Panel am1_measure_panel	sub panel of measure choices
Panel am1_button_panel	sub panel of buttons
Label am1_header_label	overall header label
CheckboxGroup am1_cbg	checkbox group - only 1 can be selected
Checkbox am1_cb	array of measure choices
Button am1_button	array of buttons

Panel am2_panel	overall panel - second measure
Panel am2_measure_panel	sub panel of measure choices
Panel am2_button_panel	sub panel of buttons
Label am2_header_label	overall header label
CheckboxGroup am2_cbg	checkbox group - only 1 can be selected
Checkbox am2_cb	array of measure choices
Button am2_button	array of buttons

The following sets up the advanced 3D scatter plot display interface screens. The panel consist of three checkbox groups (one for the x dimension of the cube, the y dimension of the cube and the z dimension):

Panel asp_panel	overall advanced scatter plot panel
Panel asp_goal_panel	sub panel containing goals to select
Panel asp_button_panel	sub panel containing the buttons
Label asp_header_label	overall panel header label
CheckboxGroup asp_cbgx	x-cube group
CheckboxGroup asp_cbgy	y-cube group
CheckboxGroup asp_cbgz	z-cube group
Checkbox asp_goalx	array of x-cube choices
Checkbox asp_goaly	array of y-cube choices
Checkbox asp_goalz	array of z-cube choices
Label xaxisa	x-cube label
Label yaxisa	y-cube label
Label zaxisa	z-cube label
Button asp_button	array of buttons

The following sets up the goal display interface screens. There are two panels, one to select the trains (gda) and one to select the goals (gdg).

Panel gda_train_panel	overall panel for train choices
Panel gda_button_panel	button panel
Label gda_header_label	overall panel header label
Checkbox gda_cb	array of train choices
Button gda_button	array of buttons

Panel gdg_panel	overall panel for goal choices
Panel gdg_goal_panel	sub panel of goal choices
Panel gdg_button_panel	sub panel of buttons
Label gdg_header_label	overall panel header label
Checkbox gdg_cb	array of goal choices
Button gdg_button	array of buttons

The following sets up the measure display interface screens. There are two panels, one to select the trains (mda) and one to select the measures (mdm).

Panel mda_panel	the overall panel
Panel mda_train_panel	sub panel of train choices
Panel mda_button_panel	sub panel of buttons
Label mda_header_label	overall header label
Checkbox mda_cb	array of train choices
Button mda_button	array of buttons

Panel mdm_panel	the overall panel
Panel mdm_measure_panel	sub panel of measure choices
Panel mdm_button_panel	sub panel of buttons

Label mdm_header_label	overall header label
Checkbox mdm_cb	array of measure choices
Button mdm_button	array of buttons

The following sets up the sensitivity analysis display interface screens. There are two panels, one to select the trains (saga) and one to assign the new weights (sag)

Panel saga_panel	the overall sag panel
Panel saga_train_panel	the sub panel of train choices
Panel saga_button_panel	the sub panel of buttons
Label saga_header_label	the overall header label
Checkbox saga_cb	array of train choices
Button saga_button	array of buttons

Panel sag_panel	overall panel of goal and weights
Panel sag_top_panel	sub panel
Panel sag_sub_panel	sub panel
Panel sag_button_panel	sub panel of buttons
CheckboxGroup sag_cbg	array of checkbox groups
Checkbox sag1_cb	array of goal choices
Checkbox sag2_cb	array of row choices
Checkbox sag3_cb	array of weight choices
Scrollbar sag_sb	scrollbar used to assign weights
Label sag_header_label	overall header label
Label sag_wt_label	weight labels
Button sag_button	array of buttons
float sag_wt_array	array containing the new weights

The following sets up the 3D scatter plot display interface screens. The panel consist of three checkbox groups (one for the x axis, the y axis and the z axis):

Panel sp_panel	the overall scatter plot panel
Panel sp_goal_panel	sub panel containing goals to select
Panel sp_button_panel	sub panel containing buttons
Label sp_header_label	overall panel header information
CheckboxGroup sp_cbgx	x-axis group
CheckboxGroup sp_cbgy	y-axis group
CheckboxGroup sp_cbgz	z-axis group
Checkbox sp_goalx	array of x-axis choices
Checkbox sp_goaly	array of y-axis choices
Checkbox sp_goalz	array of z-axis choices
Label xaxis	x-axis label
Label yaxis	y-axis label
Label zaxis	z-axis label
Button sp_button	array of buttons

The following sets up the color interface screen. This screen is used as the third screen for both the animated alternative and animated measure displays

Panel color_panel	overall screen panel
Panel color_header_panel	sub panel of header labels
Panel color_choice_panel	sub panel of color choices
Panel color_button_panel	sub panel of buttons
Label color_header_label	array of header labels
Label color_label	array of color labels
CheckboxGroup color_cbg1	checkbox group for first color
CheckboxGroup color_cbg2	checkbox group for second color
Checkbox color_cb1	array of color choices - first selection
Checkbox color_cb2	array of color choices - second selection
Button color_button	array of buttons

The following sets up the help panel screen. This consists primarily of a text area populated by the readHelpFile method

Panel help_panel	overall panel
TextArea help_text_area	text area containing the help file

The following defines the cardlayout for the interface for the screens. Once all the panels are added, it acts as a deck of cards that can be shuffled, displaying one screen at any one time

CardLayout screen_cardlayout	the layout - cardlayout
Panel screen_cards	a panel of cards using cardlayout

The following sets up the information screen panel. This consists of 8 labels. Also, a string array containing information for the scatter plots is defined here.

Panel screen_panel	overall screen information panel
Label screen_label	array of labels
String spasp_string	array of scatter plot information (x,y, z axis, etc)

The following are Instances of all the Corespace drivers used in the application

CS_GDevice GDevice	2D CoreSpace graphics driver
CS_2DDevice G2DDevice	2D CoreSpace 2d graphic function engine
CS_FSTDevice FSTriDevice	3D CoreSpace flat shaded triangle engine
CS_PolySortDevice PSDevice	3D CoreSpace polygon sorting device

METHODS:

****

Name:
action

Constructor:
boolean action (Event evt, Object arg)

Description:
This method is primarily checking for a button event.

Methods Called:
doButton

Called By:
Whenever a button is pushed

****

Name:
buildDisplayObjects

Constructor:
buildDisplayObjects ()

Description:
This method instantiates the three-dimensional objects and points to be used in each of the displays

Methods Called:
None

Called By:
newWorld

****

Name:
buildObjects

Constructor:
buildObjects ()

Description:
This method instantiates the file data array, goal object array, the measure object array, and the sag weight array, new value array, and new weight array

Methods Called:
None

Called By:
start

****

Name:
checkGoalWeight

Constructor:
checkGoalWeight(Goal goal_data[][], int goal_number, float new_weight)

Description:
This method changes the weight of a selected goal and all associated goals to ensure the sum of the weights equal one

Methods Called:
None

Called By:
Not currently being used - user interface has not been implemented

****

Name:
checkMeasureWeight

Constructor:
checkMeasureWeight(Measure measure_data[][], int measure_number, float new_weight)

Description:
This method changes the weight of a selected measure and all associated measures to ensure the sum of the weights equal one

Methods Called:
None

Called By:
Not currently being used - user interface has not been implemented

****
Name:
clearDisplay

Constructor:
clearDisplay()

Description:
This method removes all of the 3D objects from the simulation

Methods Called:
None

Called By:
doButton (whenever a display is chosen, the screen is cleared)

****
Name:
computeAbsGoal

Constructor:
float computeAbsGoal (Goal goal_data[][], int train number, int goal_number)

Description:
This method returns a number representing the absolute weight (the product of all the weights of the goals above a subgoal) of a subgoal

Methods Called:
None

Called By:
start

****
Name:
computeAbsMeasure

Constructor:
float computeAbsMeasure(Goal goal_data[][], int train_number, int measure_number)

Description:
This method returns a number representing the absolute weight (the product of all the weights of the goals above a measure) of a measure

Methods Called:
None

Called By:
updateAbsoluteWeight

****

Name:
computeGoalUtilityValue

Constructor:
float computeGoalUtilityValue(Goal goal_data[][], Measure measure_data[][], int train_number, int goal_number)

Description:
This function returns a floating point number representing the goal's value

Methods Called:
None

Called By:
updateGoalUtility

****

Name:
display3DSP

Constructor:
display3DSP(int xgoal, int ygoal, int zgoal)

Description:
This method enables the user to assign any one of the five criteria goals or the CERCLA value to the x, y, or z axis and then plots 3D cubes based on the values for the variables chosen.

Methods Called:
None

Called By:
doButton

****

Name:

displayA3DSP

Constructor:

displayA3DSP (int xgoal, int ygoal, int zgoal, int xcube, int ycube, int zcube)

Description:

This method enables the user to assign any one of the five criteria goals or the CERCLA value to the x, y, or z axis and to the height, width, and depth of the cubes, and then plots 3D cubes based on the values for the variables chosen.

Methods Called:

None

Called By:

doButton

****

Name:

displayAnimatedAlternative

Constructor:

displayAnimatedAlternative(int a1, int a2)

Description:

This method displays a cube and a sphere representing the alternatives chosen. The objects appear to travel across the screen corresponding to the values of the 28 evaluation measures. Enables the user to quickly and easily compare two alternatives.

Methods Called:

getColor

Called By:

doButton

****

Name:

displayAnimatedMeasure

Constructor:

displayAnimatedMeasure (int m1, int m2)

Description:

This method displays a cube and a sphere representing the measures chosen. The objects appear to travel across the screen corresponding to the evaluation measure values for each of the 23 alternatives. Enables the user to quickly and easily compare two measures.

Methods Called:

getColor

Called By:

doButton

****

Name:

displayGoals

Constructor:

displayGoals ()

Description:

This method enables the user to select any combination of trains and goals. The cubes displayed represent a goal's value for a given alternative. The color of the cubes are color coded to reflect their weight value.

Methods Called:

getGSelArray
getXDelta
getXStart

Called By:

doButton

****

Name:

displayHierarchy

Constructor:

displayHierarchy()

Description:

This method displays the overall hierarchy of the data. It is the first screen shown upon starting the applet.

Methods Called:

None

Called By:
start
doButton

****

Name:
displayMarkerSpheres

Constructor:
displayMarkerSpheres()

Description:
This method displays a sphere at the following points:
{0,0,0} (red), {0,0, max z} (green), {max x, 0, 0} (green),
{max x, 0, max z} (blue), {max x, max y, max z} (blue)

Methods Called:
None

Called By:
doButton

****

Name:
displayMeasures()

Constructor:
displayMeaures ()

Description:
This method enables the user to select any combination of trains and measures. The cubes displayed represent a measure's value for a given alternative. The color of the cubes are color coded to reflect their weight value.

Methods Called:
getMSelArray
getXDelta
getXStart

Called By:
doButton

****

Name:
    displaySAG

Constructor:
    displaySAG()

Description:
    This method enables a user to assign different weights to each of the five criteria goals. For a given display, a user may select up to five different weights to be displayed. When going from one row to another, if the CERCLA value decreases, the color of the cube is yellow. If the CERCLA value increases, the color is green. The best CERCLA value is blue and the worst value is red.

Methods Called:
    getChoices
    getXDelta
    getXStart

Called By:
    doButton

****

Name:
    doButton

Constructor:
    doButton (String bname)

Description:
    This method performs a variety of services based upon the button selected by the user

Methods Called:
    None

Called By:
    action

****

Name:
formatField

Constructor:
String formatField(String name, float value)

Description:
Returns a string of maximum length four, formatted as such: name: value

Methods Called:
stringFour

Called By:
showValues

****

Name:
frameAwithB

Constructor:
frameAwithB(MyObject3D base, MyObject3D frame)

Description:
Moves and rotates frame to be centered and oriented the same way as base, effectively creating a frame for it (especially if frame is wireframed)

Methods Called:
None

Called By:
showValues

****

Name:
getChoice

Constructor:
int getChoice (int choice)

Description:

Based on the integer valuepassed in, returns the user selected choice.

1 - checks asp and sp display choice for the x-axis
2 - checks asp and sp display choice for the y-axis
3 - checks asp and sp display choice for the z-axis
4 - checks asp display choice for the x-cube
5 - checks asp display choice for the y-cube
6 - checks asp display choice for the z-cube
7 - checks aa display choice for the first alternative
8 - checks aa display choice for the second alternative
9 - checks am display choice for the first measure
10 - checks am display choice for the second measure
11 - checks sag display choice for the goal selected
12 - checks sag display choice for the number of rows to be displayed

Methods Called:

None

Called By:

display3DSP
displayA3DSP
displaySAG
displayAnimatedAlternative
displayAnimatedMeasure

****

Name:

getColor

Constructor:

Color getColor (int choice)

Description:

This method returns a color object based on the user's selection

Methods Called:

None

Called By:

displayAnimatedAlternative
displayAnimatedMeasure

****

Name:
getGSelArray

Constructor:
int getGSelArray (int select_array[][])

Description:
Updates the selection array based upon the alternatives and goals chosen. Returns the number of trains selected

Methods Called:
None

Called By:
displayGoal

****

Name:
getMSelArray

Constructor:
int getMSelArray (int select_array[][])

Description:
Updates the selection array based upon the alternatives and measures chosen. Returns the number of trains selected.

Methods Called:
None

Called By:
displayMeasure

****

Name:
getXDelta

Constructor:
float getXDelta (int num_trains)

Description:
returns the spacing for the x axis, based on the number of trains selected.

Methods Called:
    None

Called By:
    displayGoal
    displayMeasure
    displaySAG

****

Name:
    getXStart

Constructor:
    float getXStart (int num_trains)

Description:
    returns the starting position along the x axis, based on the number of trains selected.

Methods Called:
    None

Called By:
    displayGoal
    displayMeasure
    displaySAG

****

Name:
    handleEvent

Constructor:
    boolean handleEvent (Event evt)

Description:
    This method is called whenever an event occurs. Looks specifically for instances of a scrollbar event. Updates the new_weight_array based upon user input.

Methods Called:
    None

Called By:
    Whenever a scrollbar event occurs

****

Name:
init

Constructor:
init ()

Description:
Override this method to build all the interface screen panels.

Methods Called:
None

Called By:
Initialization occurs when the applet is first loaded.

****

Name:
keyDown

Constructor:
boolean keyDown (Event event, int key)

Description:
This methods checks to see what key was selected, then performs the required action.

Methods Called:
None

Called By:
Whenever a keyboard key is pressed

****

Name:
loadFileData

Constructor:
loadFileData (float file_data[][], Measure measure_data[][], Goal goal_data[][])

Description:
This method loads the data into the measure and goal objects

Methods Called:

Measure.setRawUtility
Measure.setWeight
Goal.setWeight

Called By:

start

****

Name:

mouseDown

Constructor:

boolean mouseDown (Event evt, int x, int y)

Description:

This method is checking if an object is grapped and captures the x and y coordinates.

Methods Called:

showValues

Called By:

Whenever a mouse down event occurs

****

Name:

mouseUp

Constructor:

boolean MouseUp (Event evt, int x, int y)

Description:

This method removes the outline cube from the simulation

Methods Called:

None

Called By:

Whenever the mouse button is released

****

Name:
newWorld

Constructor:
newWorld ()

Description:
This method builds the three-dimensional world. A three-dimensional world must be built before any other objects are placed into the simulation.

Methods Called:
buildDisplayObjects

Called By:
start

****

Name:
paint

Constructor:
paint (Graphics g)

Description:

Override the paint method. Can check to make sure world is not set to null.

Methods Called:
None

Called By:
Whenever something is drawn on the screen

****

Name:
pause

Constructor:
pause ()

Description:
This method pauses all activity on the screen.

Methods Called:
None

Called By:
User interaction - key F7

****

Name:
processNavButton

Constructor:
processNavButton()

Description:
This method changes the view based upon user input (rotate, move left, up, etc).

Methods Called:
None

Called By:
run

****

Name:
readDataFile

Constructor:
readDataFile (float file_data[][], String file_name)

Description:
This method opens an input stream to read the data file containing the raw measure scores and goal and measure weights. The data is read into the file data array.

Methods Called:
None

Called By:
start

****

Name:
readHelpFile

Constructor:
readHelpFile (String file_name, TextArea help_ta)

Description:
This method reads the help file data into a text area which can then be displayed to the user

Methods Called:
None

Called By:
start

****

Name:
run

Constructor:
run ()

Description:
This method is always "running" checking for a navigational button being pressed. Also checks if animated alternatives or animated measures was selected and moves the cubes and spheres across the screeen.

Methods Called:
None

Called By:
always running

****

Name:
showValues

Constructor:
showValues()

Description:
This method, based on the currently selected display, updates object information into the appropriate screen labels.

Methods Called:
  formatField

Called By:
  mouseDown

****

Name:
  start

Constructor:
  start()

Description:
  This method calls several other methods to "start" the applet

Methods Called:
  buildObjects
  readDataFile
  readHelpFile
  loadFileData
  updateMeasureUtility
  updateWeightedMeasureUtility
  updateGoalUtility
  updateAbsoluteWeights
  NewWorld
  clearDisplay
  displayHierarchy

Called By:
  After an applet is initialized, it is automatically started

****

Name:
  stop

Constructor:
  stop()

Description:
  This methods sets all objects equal to null

Methods Called:
  None

Called By:
Stopping occurs when the reader leaves the page that contains a currently running applet.

****

Name:
stringFour

Constructor:
String stringFour(float value, int max_chars)

Description:
Returns a string which corresponds to value but has length of at most four characters.

Methods Called:
None

Called By:
formatField

****

Name:
update

Constructor:
update (Graphics g)

Description:
include the update function to override internal screen clearing

Methods Called:
paint

Called By:
Whenever clearing the screen

****

Name:
updateAbsoluteWeight

Constructor:
updateAbsoluteWeight (Measure measure_data[][], Goal goal_data[][])

Description:
This method has each measure and each goal update their absolute weight and their absolute weighted value

Methods Called:
computeAbsMeasure
computeAbsGoal
Measure.updateAbsWeightValue
Measure.updateAbsUtilityValue
Goal.updateAbsWeightValue
Goal.updateAbsUtilityValue

Called By:
start

****

Name:
updateGoalUtility

Constructor:
updateGoalUtility(Goal goal_data[][], Measure measure_data[][])

Description:
This method has each goal update its value

Methods Called:
computeGoalUtilityValue
Goal.updateUtilityValue

Called By:
start

****

Name:
updateMeasureUtility

Constructor:
updateMeasureUtility (Measure measure_data[][])

Description:
This method loops through all the measures and has each measure update its value based upon its raw score

Methods Called:
Measure.computeUtilityValue

Called By:
start

****

Name:
updateWeightedMeasureUtility

Constructor:
updateWeightedMeasureUtility (Measure measure_data[][])

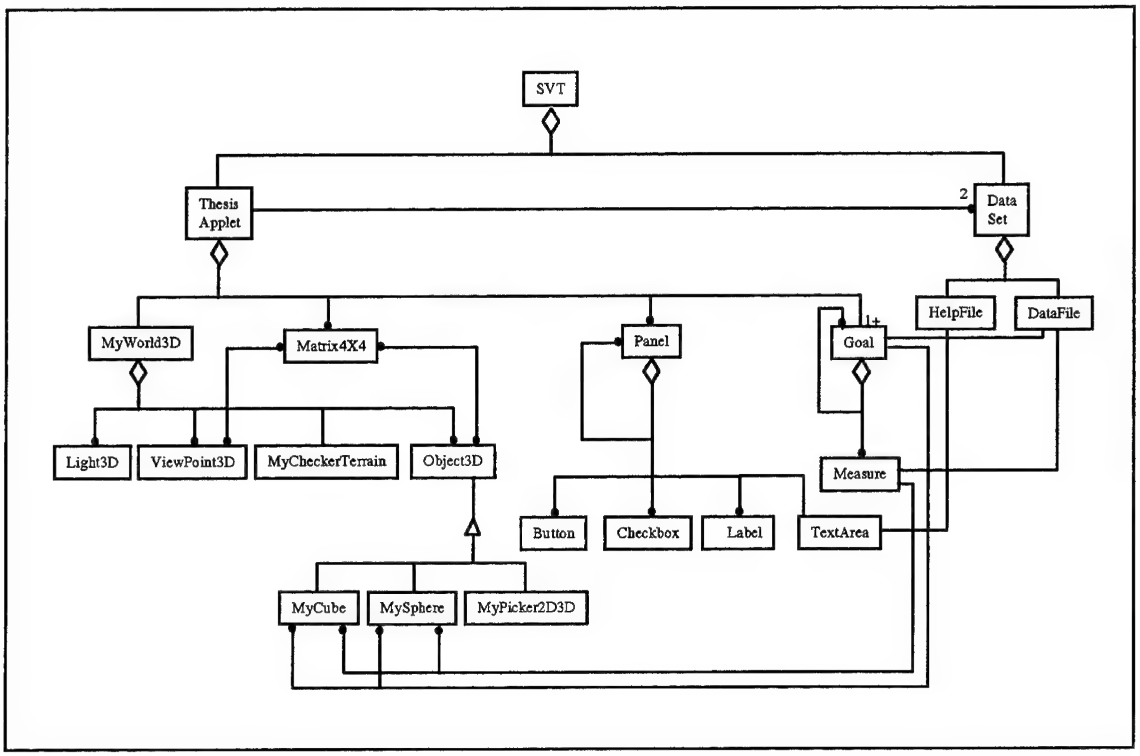
Description:
This method computes the weighted value for each measure.

Methods Called:
Measure.updateWtUtilityValue

Called By:
start

Appendix F: Software Object Model

The following model represents the software visualization tool (SVT). Appendix C contains a complete description of the variables and methods contained in each class.



Appendix G: CERCLA Data

Train	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
Train 1	2	10	0	1	1	90	100	1	3	1	0	0	90	100
Train 2	2	70	948.87	2	2	30	97.84	2	2	2	0	0	26.45	97.84
Train 3	2	70	0	2	1	30	100	2	2	2	0	0	27.73	100
Train 4	2	70	0	2	1	30	100	2	2	2	0	0	23.95	100
Train 5	2	10	0	3	1	90	100	1	2	3	0	0	90	100
Train 6	1	10	0	1	1	90	100	2	3	2	0	0	90	100
Train 7	1	10	0	1	1	90	100	2	3	1	0	0	90	100
Train 8	1	10	0	3	1	90	100	1	2	3	0	0	90	100
Train 9	1	10	17385.7	1	1	90	60.42	1	2	1	0	0	90	60.42
Train 10	2	70	0	2	1	30	100	2	3	2	0	0	23.83	100
Train 11	2	10	0	3	1	90	100	1	3	3	0	0	90	100
Train 12	1	10	0	1	1	90	100	2	3	2	0	0	90	100
Train 13	1	10	0	1	1	90	100	2	3	1	0	0	90	100
Train 14	1	10	0	3	1	90	100	1	3	3	0	0	90	100
Train 15	1	10	32901.54	1	2	90	25.09	1	3	1	0	0	90	25.09
Train 16	2	70	0	2	1	30	100	1	2	2	0	0	22.6	100
Train 17	2	10	0	3	1	90	100	1	2	3	0	0	90	100
Train 18	1	10	0	1	1	90	100	1	3	2	0	0	90	100
Train 19	1	10	0	1	1	90	100	1	3	1	0	0	90	100
Train 20	1	10	0	3	1	90	100	1	2	3	0	0	90	100
Train 21	1	10	15240.91	1	1	90	65.3	1	2	1	0	0	90	65.3
Train 22	1	70	948.87	1	4	30	97.84	2	2	2	0	0	0	97.84
Train 23	1	10	34760.67	1	2	90	20.86	1	3	1	0	0	90	20.86
M Weights	0.1	0.7	0.2	1	1	0.75	0.25	0.333	0.25	0.083	0.25	0.083	0.75	0.25
G Weights	1	0.25	0.25	0.167	0.167	0.167	0.5	0.5	0.25	0.25	0.25	0.25	0.25	0.25

Train	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28
Train 1	3	5	4	4	2003.33	2	10	2	2	5	4	1	2	4661241
Train 2	3	3	5	6	2032	1	3	10	3	2	1	1	1	19874448
Train 3	3	4	4	7	2032	1	5	10	0	2	1	1	1	19986501
Train 4	3	4	4	7	2032	1	8	10	0	2	1	1	1	32113745
Train 5	3	4	4	7	2007.7	2	10	3	1	2	2	1	4	2697842
Train 6	3	4	4	2	2007.7	2	12	3	0	4	3	1	2	10430921
Train 7	3	4	4	3	2007.7	2	12	3	0	3	3	1	1	9255230
Train 8	3	4	4	1	2007.7	1	8	0	6	2	3	1	4	7395690
Train 9	3	4	4	4	2009.25	1	7	3	4	2	3	1	1	4030652
Train 10	3	5	4	7	2032	2	13	4	0	6	5	1	3	14804185
Train 11	3	5	4	7	2003.7	3	14	3	2	6	5	1	4	2159602
Train 12	3	5	4	2	2003.12	3	16	3	1	6	5	1	3	9759951
Train 13	3	5	4	3	2003.12	3	16	3	1	6	5	1	3	8606073
Train 14	3	5	4	1	2003.7	3	12	0	7	6	5	1	4	6770288
Train 15	3	5	5	4	2004.26	2	11	3	5	6	5	1	3	3553454
Train 16	3	5	4	7	2032	2	9	1	1	3	5	1	3	13209947
Train 17	4	4	7	7	2009.72	3	10	1	4	3	5	1	4	1228989
Train 18	3	5	4	2	2009.14	3	12	1	3	4	5	1	3	7913753
Train 19	3	5	4	3	2009.14	3	12	1	3	3	5	1	3	6930784
Train 20	4	3	7	1	2009.72	3	8	0	9	3	5	1	4	5156755
Train 21	4	3	7	4	2010.28	2	7	1	7	3	5	1	3	2416388
Train 22	1	2	7	7	2032	0	0	10	5	0	0	1	1	1228989
Train 23	3	5	4	4	2003.83	2	8	2	4	3	5	1	1	3629236
M Weights	1	1	0.5	0.5	1	0.5	0.5	1	1	1	1	1	1	1
G Weights	0.25	0.25	0.333	0.333	0.333	0.25	0.25	0.25	0.25	0.5	0.5			

Bibliography

Asimov, Daniel. "The Grand Tour: Animating and Interacting with Projections of the Data"., Proceedings Visualization '95: 407-408 (October/November 1995).

Booker, Ellis. "A New Method of Seeing?", ComputerWorld, 27: 28 (5 Apr 1993).

Brittain, Donald L. and others. "Design of an End-User Data Visualization System", Proceedings Visualization '90: 323-327 (October 1990).

Brown, Judith R. and others. Visualization. New York: John Wiley and Sons, Inc., 1995.

Brown, Steven D. "Technical Graphics with a Twist: Axum", Analytical Chemistry, 66: 95A-96A (15 January 1994).

Chi, Ed Huai-hsin and others. "Flexible Information Visualization of Multivariate Data from Biological Sequence Similarity Searches", Proceedings Visualization '96: 133-140 (October/November 1996).

Cleveland, William S. "A Model for Studying Display Methods", Proceedings Visualization '93: 351-352 (October 1993).

Deckro, Richard F., Jack M. Kloeber, Jr., and Jack A. Jackson. "Communicating Post Optimality Analysis: Visualizing Sensitivity Analysis", Proceedings of the 27th Annual Westen Decision Science Meeting: (April 1988).

Douglas, John. "Visualizing Complex Systems", EPRI Journal, 19: 18-23 (December 1994).

Eick, Stephen G. and Daniel E. Fyock. "Visualizing Corporate Data", AT&T Technical Journal, 75: 74-85 (January 1996).

Ellison, Aaron M. "Right Between the Eyes - Visualizing Data by William S. Cleveland", Bioscience, 44: 622-623 (October 1994).

Flanagan, David. Java in a Nutshell (Second Edition). Sebastopol, CA: O'Reilly, 1997.

Fortner, Brand. The Data Handboook (Second Edition). Santa Clara, CA: Telos (Springer-Verlay Publishers), 1995.

Fracchia, David. "Toward Image Understanding", Proceedings Visualization '96: 442 (October/November 1996).

Gershon, Nahum. "Breaking the Myth: One Picture is Not (Always) Worth a Thousand Words", Proceedings Visualization '96: 441-443 (October/November 1996).

Gershon, Nahum and Jeff Dozier. "The Difficulty With Data", Byte, 18: 143-148 (April 1993).

Gershon, Nahum and others. "Is Visualization Really Necessary? The Role of Visualization in Science, Engineering, and Medicine", Proceedings Visualization '93: 343-346 (October 1993).

Gershon, Nahum D. "A Picture is Not a Picture is Not a Picture . . . A Picture Could be Worth a 1,000, 1/1000, or -1,000 Words", Proceedings Visualization '96: 441 (October/November 1996).

Glassner, Andrew. "Different Media Means Different Messages", Proceedings Visualization '96: 442 (October/November 1996).

Globus, Al and others. "Validation, Verification and Evaluation", Proceedings Visualization '94: 414-418 (October 1994).

Gosling, James. The Java Programming Language. Reading, MA: Addison Wesley, 1996.

Grinstein, Georges. "Extending Visualization from the Pixel to Icons", Proceedings Visualization '95: 405-406 (October/November 1995).

Hanson, Andrew J., Tamara Munzner, and George Francis "Interactive Methods for Visualizable Geometry", Computer (IEEE): 73-83 (July 1994).

Heath, Michael T., Allen D. Murphy, and Diane T. Rover. "The Visual Display of Parallel Performance Data", Computer (IEEE), 28: 30-36 (November 1995).

Inselberg, Alfred. "Parallel Coordinates - Visualization of N-Dimensional Geometry", Proceedings Visualization '95: 408-409 (October/November 1995).

Inselberg, Alfred. "Visualizing Multidimensional (Multivariate Data) and Relations – Perception vs Geometry", Proceedings Visualization '95: 405-411 (October/November 1995).

Kaufman, Arie E. "Visualization", Computer (IEEE), 27: 18-19 (July 1994).

Keller, Peter R. and Mary M. Keller. Visual Cues: Practical Data Visualization. Los Alamitos, CA: IEEE Computer Society Press and Piscataway, NJ: IEEE Press, 1993.

Kerchus, Christine E., A CERCLA-Based Decision Model to Support Remedy Selection for an Uncertain Volume of Contaminants at a DOE Facility. Thesis, Masters of Science, Air Force Institute of Technology, AFIT/GEE/GOR/97D-01. 1997.

Kohn, Marek. "Is This the End of Abstract Though?", New Scientist, 43: 36-39 (17 September 1994).

LeMay, Laura and Charles L. Perkins. Teach Yourself Java 1.1 in 21 Days (Second Edition). Indianapolis, IN: Sams.net, 1997.

Levkowitz, Haim. "Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters", Proceedings Visualization '91: 164-170 (October 1991).

Levkowitz, Haim. "Color vs. Black-and-White in Visualization", Proceedings Visualization '91: 336-339 (October 1991).

Logical Decisions. Logical Decisions for Windows. Golden, CO: Logical Decisions, 1995.

Marchak, Frank M. "Perceptual Effectiveness of Visualization Techniques", Proceedings Visualization '93: 352 (October 1993).

Marchak, Frank M. and others. "The Psychology of Visualization", Proceedings Visualization '93: 352-354 (October 1993).

Martin, James. "Beyond Pie Charts and Spreadsheets", Computerworld, 30: 37 (27 May 1996).

McKim, Robert H. Experiences in Visual Thinking (Second Edition). Belmont, CA: Wadsworth, Inc, 1980.

Menoher, Paul E., Jr. "Fielding New Systems for Today's Army and Army XXI", Army, 46: 135-140 (October 1996).

Meyer, Gary W. "Some Problems with Color as a Visualization Tool", Proceedings Visualization '91: 338 (October 1991).

Mihalisin, T., J. Timlin, and J. Schwegler. "Visualization and Analysis of Multivariate Data: A Technique for All Fields", Proceedings Visualization '91: 171- 178 (October 1991).

Mones-Hattal, Barbara. "A Picture May (or May Not Be) Worth a Thousand Words: Lessons From the WWW?", Proceedings Visualization '96: 442 (October/November 1996).

Naughton, Patrick. The Java Handbook. Berkely, CA: Osborne McGraw-Hill, 1996.

Pressman, Roger S. Software Engineering, A Practitioner's Approach (Third Edition). New York: McGraw-Hill, Inc., 1992.

Reuter, Laurie. "Human Perception and Visualization", Proceedings Visualization '90: 401-406 (October 1990).

Rhyne, Theresa-Marie. "Scientific Visualization and Technology Transfer", Computer (IEEE), 28: 94-96 (July 1995).

Ribarsky, William and others. "Glyphmaker: Creating Customized Visualizations of Complex Data", Computer (IEEE), 27: 57-64 (July 1994).

Ribarsky, William. "Navigating the Data Flood", Byte, 18: 129-135 (April 1993).

Richman, Dan. "Your Data Ought to be in Pictures", Computerworld, 30: 52 (10 June 1996).

Robertson, George G., Stuart K. Card, and Jock D. MacKinlay. "Information Visualization Using 3D Interactive Information", Communications of the ACM, 36: 56-71 (April 1993).

Robertson, Philip K. "Color by Design", Proceedings Visualization '91: 338-339 (October 1991).

Rogowitz, Bernice E. "The Effects of Mapping Choices on Data Perception and Interpretation", Proceedings Visualization '93: 352 (October 1993).

Rose, Russ. "P1000: A Picture is Worth 1000 Words", Proceedings Visualization '96: 442 (October/November 1996).

Rumbaugh, James and others. Object-Oriented Modeling and Design. Englewood Cliffs, NJ: Prentice Hall, 1991.

Rushmeir, Holly and others. "Metrics and Benchmarks for Visualization", Proceedings Visualization '95: 422-426 (October/November 1995).

Samtaney, Ravi and others. "Visualizing Features and Tracking Their Evolution", Computer (IEEE): 20-26 (July 1994).

Sommerville, Ian. Software Engineering (Fourth Edition). New York: Addison-Wesley Publishing Company, 1992.

Teresko, John. "Too Much Data, Too Little Information", Industry Week, 245: 66-70 (19 August 1996).

Treinish, Lloyd A. "Inside Multidimensional Data", Byte, 18: 132 - 133 (April 1993).

Tufte, Edward. Envisioning Information (Second Edition). Cheshire, CN: Graphics Press, 1991.

Tufte, Edward. Visual Display of Information. Cheshire, CN: Graphics Press, 1983.

Tufte, Edward. Visual Explanations. Cheshire, CN: Graphics Press, 1997.

Tukey, Paul. "Visual Perception and Scientific Data Display", Proceedings Visualization '90: 401-403 (October 1990).

Uselton, Sam and others. "Validation, Verification and Evaluation", Proceedings Visualization '94: 414-418 (October 1994).

Van Dam, Laura. "A Picture is Worth 1,000 Numbers", MIT's Technology Review, 95: 1-8 (1 May 1992).

Wayner, Peter. "Image Building", Byte, 18: 137-141 (April 1993).

Ward, Matthew O. "Xmdv Tool: Integrating Multiple Methods for Visualizing Multivariate Data", Proceedings Visualization '94: 326-331 (October 1994).

Weber, Jack. "Visualization: Seeing is Believing", Byte, 18: 120-128 (April 1993).

Yrastorza, Teresa. "The Big Picture", Computerworld: F9-F11 (August 1996)

Vita

Captain Steven C. Oimoen was born on 30 January 1962 in Monroe, WI. He graduated from Monona Grove High School in Monona, WI in 1980. He enlisted in the United States Air Force in March 1982. He received an Associate in Applied Science in Personnel Administration and an Associate in Applied Science in Computer Science Technology from the Community College of the Air Force in March 1990. He graduated with a Bachelor of Science in Computer Science and a Bachelor of Arts in Mathematics from Hawaii Pacific University in May 1993. He entered Officer Training School (OTS) in October 1993 and was commissioned in January 1994.

His first assignment was at Luke AFB, Arizona as an administrative specialist, followed by an assignment to Randolph AFB, TX. In January 1988, he retrained into the computer programming career field. As a programmer, he was stationed at Air Force Global Weather Central, Offutt AFB, NE and the 1854th Command, Control, and Intelligence Support Squadron, Hickam AFB, HI. Upon completing OTS, Captain Oimoen served as a software engineer, instructing at Keesler AFB, Mississippi. In August 1996, he entered the School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, OH.

Permanent Address: 4908 Shore Acres Rd
Monona, WI 53716

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1998	Master's Thesis

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
AN ADVANCED VISUALIZATION METHOD FOR AN OPERATIONS RESEARCH ANALYSIS	
6. AUTHOR(S) Steven C. Oimoen, Captain, USAF	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology 2750 P Street WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GOR/ENG/98M-01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution unlimited	

13. ABSTRACT *(Maximum 200 words)*

Visualizing multidimensional data using only two dimensions and conventional visualization techniques limits the understanding of the data set. Underlying structures or patterns within the data can easily go unnoticed. In order to gain additional insight into an analysis, incorporation of visualization and multidimensional graphics into the analysis results should be accomplished. The results must ensure that the information portrayed is not misleading or misunderstood. The integrity of the data must be preserved throughout the transformation. The prmary objective of this research effort is to identify techniques to visualize multidimensional data and then develop a software tool to display the multidimensional data in order to gain insight into the data and analysis performed.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Visualization, Information Visualization, Multivariate Data, Multidimensional Data, Decision Analysis, Computer Programming, Object-Oriented Programming	220
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	**PR** - Project
G - Grant	**TA** - Task
PE - Program Element	**WU** - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with....; Trans. of....; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.